

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA ELECTRÓNICA

Trabajo de titulación previo a la obtención del título de:
INGENIERA ELECTRÓNICA

TEMA:
ALMACÉN INTELIGENTE MEDIANTE ROBOT OMNIDIRECCIONAL Y
VISIÓN ARTIFICIAL

AUTORA:
ZAIRA BELÉN CALVOPÍÑA LIMAICO

TUTORA:
CARMEN JOHANNA CELI SÁNCHEZ

Quito, noviembre de 2018

CESIÓN DE DERECHOS DE AUTOR

Yo Zaira Belén Calvopiña Limaico, con documentos de identificación N° 1721092755, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autora del trabajo de titulación intitulado: “ALMACÉN INTELIGENTE MEDIANTE ROBOT OMNIDIRECCIONAL Y VISIÓN ARTIFICIAL”, mismo que ha sido desarrollado para optar por el título de Ingeniera Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autora me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Zaira Belén Calvopiña Limaico

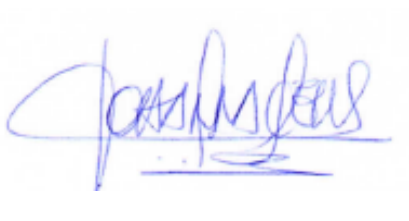
CI: 172109275-5

Quito, noviembre 2018.

DECLARATORIA DE COAUTORÍA DE LA DOCENTE TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Trabajo de Titulación, “ALMACÉN INTELIGENTE MEDIANTE ROBOT OMNIDIRECCIONAL Y VISIÓN ARTIFICIAL”, realizado por Zaira Belén Calvopiña Limaico, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, noviembre de 2018.



Carmen Johanna Celi Sánchez

CI: 171743780-8

DEDICATORIA

Dedico este proyecto de titulación a los pilares fundamentales de mi vida, mis padres: César Calvopiña y Susana Limaico, quienes con su trabajo, esfuerzo, dedicación y perseverancia me guían día a día en todas mis actividades y quienes me apoyaron infinitamente a lo largo de este camino, confiaron en mí y nunca dejaron que me rinda pese a cualquier dificultad.

Dedico también este proyecto a mis hermanos: Paulina, Gissela y Franklin por ser mi ejemplo de lucha, mi apoyo y por estar siempre conmigo en los buenos y malos momentos.

A mis sobrinos: César, Valeria, Danny, Victoria y Martina por ser mi motivación diaria para salir adelante y de esta forma ser un ejemplo de superación para ellos.

A mi novio, William Velasco por su amor, paciencia y por siempre estar a mi lado apoyándome y dándome ánimo con palabras de cariño y afecto.

AGRADECIMIENTO

Agradezco a Dios, por permitirme culminar mi carrera satisfactoriamente y cumplir así, mi gran sueño. A mis padres por su ejemplo de lucha, perseverancia, esfuerzo y apoyo brindado para alcanzar mi objetivo.

A mi tutora la Ing. Carmen Johanna Celi Sánchez MSc. Por su paciencia pero sobre todo por brindarme su apoyo incondicional y guiarme con sus conocimientos en la realización de este proyecto de titulación.

A la Universidad Politécnica Salesiana, cuyas instalaciones fueron por algunos años mi segunda casa, en la cual aprendí no solo a ser una profesional a carta cabal sino también una excelente persona.

Al Ing. Ricardo Zapata y a mi amiga Diana Solórzano por compartir conmigo sus conocimientos y guiarme en la realización de este proyecto.

A mi gran amigo Jonathan Aguilar por su paciencia y apoyo durante toda la carrera y por su valiosa ayuda para culminar este proyecto.

A los docentes que me guiaron en cada nivel a lo largo de la carrera y como no agradecer a mis amigos que siempre fueron un gran apoyo.

ÍNDICE DE CONTENIDO

CESIÓN DE DERECHOS DE AUTOR.....	ii
DECLARATORIA DE COAUTORÍA DE LA DOCENTE TUTORA	iii
DEDICATORIA.....	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDO.....	vi
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS	xii
RESUMEN.....	xiii
ABSTRACT	xiv
INTRODUCCIÓN.....	xiv
CAPÍTULO 1	1
ANTECEDENTES	1
1.1 Planteamiento del problema.....	1
1.2 Justificación del problema	1
1.3 Objetivos.....	1
1.3.1 Objetivo General.....	1
1.3.2 Objetivos Específicos.....	2
1.4 Propuesta de solución	2
1.5 Alcance	2
CAPÍTULO 2	3
MARCO TEÓRICO.....	3
2.1 Robotino.....	3
2.1.1 Aplicaciones de los robots móviles.....	3
2.1.2 Tareas en la industria.....	4
2.2 Códigos QR.....	4

2.2.1 Descripción de los códigos QR.....	5
2.3 Raspberry Pi.....	7
2.3.1 Descripción de la placa.....	7
2.3.2 Hardware de Raspberry Pi.....	9
2.3.3 Software de Raspberry Pi	10
2.4 Visión Artificial.....	11
2.5 Manipulación de productos.....	11
2.5.1 Producto Químico	11
2.5.2 Vías de entrada del contaminante al cuerpo	14
2.6 Elementos Electrónicos.....	15
CAPÍTULO 3	21
DISEÑO, DESARROLLO E IMPLEMENTACIÓN	21
3.1 Caracterización Robot Robotino	21
3.1.1 Chasis	21
3.1.2 Baterías	22
3.1.3 Unidad de accionamiento (Omnidrive)	22
3.1.4 Unidad de control.....	23
3.1.5 Teclado y Display	24
3.1.6 Sensores.....	25
3.1.7 Interfaces	26
3.2 Configuración de la placa Raspberry Pi 3	27
3.2.1 Inicialización del Sistema Operativo	27
3.2.2 Iniciación de Apache y PHP.	27
3.3 Desarrollo del Sistema de Lectura e Interpretación de Códigos QR.....	28
3.3.1 Generación de Códigos QR	29
3.4 Diseño del Sistema Electrónico.....	31

3.4.1 Circuito de control para eje móvil en “z”	31
3.4.2 Circuito de control pinza sujetadora	33
3.4.3 Circuito de control para lectura e interpretación de códigos QR.....	35
3.4.4 Diagrama esquemático de la placa de opto acopladores	37
3.4.5 Diagrama de Flujo del Sistema de Almacenamiento de productos.	39
3.4.6 Diagrama de Flujo del Sistema de Despacho de productos.....	40
3.4.7 Interfaz en Labview para la puesta en marcha del sistema	42
3.4.8 Resultados de la Implementación Final del Sistema en el Robotino	43
CAPÍTULO 4	46
PRUEBAS Y RESULTADOS	46
4.1 Pruebas de Hardware	46
4.2 Pruebas de Software	47
4.2.1 Prueba de lectura e interpretación de códigos QR	47
4.2.2 Prueba de ascenso y descenso del eje móvil en “z”	47
4.2.3 Prueba de apertura y cierre de pinza sujetadora	47
4.3 Pruebas en pista	48
4.3.1 Corrección de Línea y Corrección de Posición	49
4.3.2 Tiempos de Almacenamiento y Despacho	51
CONCLUSIONES	53
RECOMENDACIONES	56
CITAS BIBLIOGRÁFICAS	57
ANEXO I.....	60
ANEXO II	61
ANEXO III	62
ANEXO IV	63
ANEXO V	64

ANEXO VI.....	65
---------------	----

ÍNDICE DE FIGURAS

Figura 2.1: Robot Robotino	3
Figura 2.2: Desarrollo del Código QR	4
Figura 2.3: Elementos de un Código QR	7
Figura 2.4: Tarjeta de Desarrollo ARM	8
Figura 2.5: Pines Raspberry Pi.	10
Figura 2.6: Vías de entrada del contaminante al cuerpo	15
Figura 2.7: Motor a pasos DYNASYN	15
Figura 2.8: Easy Driver v4.4	16
Figura 2.9: Módulo LM2596 Convertidor de Voltaje DC-DC Buck	17
Figura 2.10: Sensor de Campo Magnético	18
Figura 2.11: Servomotor Hitec HS-311	19
Figura 2.12: Cámara Web Logitech C170	20
Figura 3.1: Chasis	21
Figura 3.2: Unidad de accionamiento	22
Figura 3.3: Unidad de Control	23
Figura 3.4: Interfaces de la Unidad de Control	24
Figura 3.5: Teclado y display	25
Figura 3.6: Interfaces E/S	26
Figura 3.7: Proceso de instalación de Apache	28
Figura 3.8: Interfaz del Generador de Códigos QR.	29
Figura 3.9: Opciones para generar el código QR	30
Figura 3.10: Código QR – ARSENICO	30
Figura 3.11: Interfaces E/S utilizadas de Robotino	31
Figura 3.12: Diagrama de Bloques para el control del eje móvil en “z”	32
Figura 3.13: Esquema gráfico de conexión gráfica del eje móvil en “z”	33
Figura 3.14: Diagrama de Bloques para las conexiones de la pinza sujetadora	34
Figura 3.15: Esquema gráfico de conexión gráfica para pinza sujetadora	34
Figura 3.16: Diagrama de Bloques para la Lectura e Interpretación de Códigos QR	35
Figura 3.17: Diagrama esquemático de la placa de opto acopladores	37
Figura 3.18: Diseño 3D de la placa de opto acopladores	38

Figura 3.19: Conexión total de los elementos de control al robot	38
Figura 3.20: Diagrama de flujo del proceso de almacenamiento de productos.....	39
Figura 3.21: Diagrama de flujo del proceso de despacho de productos.	40
Figura 3.22: Subproceso del diagrama de flujo del almacenamiento y despacho de productos.	41
Figura 3.23: Interfaz gráfica para el control general de Robotino.....	42
Figura 3.24: Robot compacto con los circuitos electrónicos	43
Figura 3.25: Ambiente estructurado y almacén.....	44
Figura 3.26: Caja de madera con código QR	44
Figura 3.27: Tarjeta para el despacho de productos	45
Figura 4.1: Conexiones electrónicas realizadas en el robot.....	46
Figura 4.2: Pista para desplazamiento del robot.....	48
Figura 4.3: Almacén y Robotino listos para la puesta en marcha del sistema	49

ÍNDICE DE TABLAS

Tabla 2.1: Especificaciones de la cantidad de caracteres.....	5
Tabla 2.2: Niveles de corrección de códigos QR	6
Tabla 2.3: Elementos principales de la placa Raspberry Pi 3	9
Tabla 2.4: Descripción de sustancias nocivas para el ser humano	12
Tabla 2.5: Características del motor a pasos DYNASYN	16
Tabla 2.6: Especificaciones técnicas de la cámara Logitech	20
Tabla 3.1: Valores Técnicos del Motor a pasos DC.....	23
Tabla 3.2: Cadena de bits por producto para el almacenamiento y despacho.....	36
Tabla 4.1: Resultados obtenidos en la lectura de códigos QR.....	47
Tabla 4.2: Resultados de la corrección de línea y corrección de posición por producto.....	50
Tabla 4.3: Datos obtenidos en el desplazamiento de Robotino.....	51
Tabla 4.4: Datos de Tiempo de Almacenamiento y Despacho de Productos	52

RESUMEN

En este trabajo de titulación se presenta el desarrollo de un almacén inteligente mediante un robot omnidireccional y visión artificial. Se utiliza el Robotino de la marca FESTO, que será el encargado de almacenar y despachar productos, mediante el control de un eje de libertad en “z” implementado en el robot consu pinza sujetadora.

La tarjeta de adquisición de datos Raspberry Pi 3, se usa para realizar la captura, detección y decodificación de códigos QR, así como también control del sistema de almacenamiento y despacho. Se utiliza Python para el desarrollo de los algoritmos de control del sistema de lectura e interpretación de códigos QR, éstos son los encargados de informar al robot el nombre del producto y la posición en la que se debe ubicar cada uno tanto en el almacenamiento como en el despacho.

Se realizó también el desarrollo de algoritmos de control para el movimiento del eje móvil en “z” y la pinza sujetadora.

Se desarrolla una interfaz, la cual muestra el estado, vacío o lleno, de las localidades en el almacén tridimensional. Para el desplazamiento del robot en el área de trabajo, se utiliza Labview 2015 con sus librerías específicas de control para Robotino. Se implementa el desplazamiento del robot mediante un algoritmo seguidor de línea, permitiendo que el robot se mueva de forma guiada hacia cada una de las localidades del almacén. Finalmente cabe destacar que todo el sistema implementado en el robot es totalmente autónomo.

ABSTRACT

In this titling work, the development of an intelligent warehouse using an omnidirectional robot and artificial vision is presented. The Robotino of the FESTO brand is used, which will be responsible for storing and dispatching products, by controlling an axis of freedom in "z" implemented in the robot with its gripper clamp.

The Raspberry Pi 3 data acquisition card is used to capture, detect and decode QR codes, as well as control the storage and dispatch system. Python is used for the development of control algorithms for reading and interpretation of QR codes, these are responsible for informing the robot the name of the product and the position in which each must be located both in storage and in the office.

The development of control algorithms for the movement of the movable axis in "z" and the gripper clamp was also carried out.

An interface is developed, which shows the state, empty or full, of the locations in the three-dimensional warehouse. For the movement of the robot in the work area, Labview 2015 is used with its specific control libraries for Robotino. The displacement of the robot is implemented by means of a line follower algorithm, allowing the robot to move in a guided manner towards each of the locations of the warehouse. Finally it should be noted that the entire system implemented in the robot is completely autonomous.

INTRODUCCIÓN

La manipulación de productos químicos ocasiona un peligro para los seres vivos. Es de gran importancia saber cómo operarlos para evitar que dichos riesgos generen accidentes.

Las sustancias peligrosas para la salud son aquellas que pueden causar contusiones ingresando al cuerpo por una o varias vías a la vez. Una única sustancia puede generar contusiones en varias zonas del cuerpo humano. (Universidad Autónoma de Occidente, 2011)

Este proyecto de titulación surge de la idea de evitar accidentes laborales por manipulación de sustancias nocivas, implementando un sistema de almacenamiento y despacho de productos utilizando un robot omnidireccional que al ser una máquina no sufre las consecuencias que este trabajo ocasiona en el ser humano, y una tarjeta de desarrollo Raspberry Pi 3 como el cerebro principal de este trabajo.

El proyecto de titulación abarca los siguientes capítulos:

En el capítulo 1, se muestra el planteamiento del problema, justificación, objetivos, propuesta de solución y el alcance del proyecto de titulación.

En el capítulo 2, se detalla netamente la parte teórica, es decir, los datos técnicos y características propias de Robotino y de la tarjeta desarrollo Raspberry Pi 3.

En el capítulo 3, se describe el desarrollo del proceso realizado para el almacenamiento y despacho de productos. Se inicia desde la creación de códigos QR, algoritmos de captura, detección y decodificación de los mismos, seguido de la creación de algoritmos de control del eje móvil en “z” y pinza sujetadora, además del manejo del sensor magnético y finalmente la creación de algoritmos de control para el desplazamiento del robot en un ambiente estructurado.

En el capítulo 4, se presentan las pruebas realizadas para la validación del proyecto de titulación, las cuales están fundamentadas en los siguientes aspectos: tiempos de almacenamiento y despacho de cada producto, cantidad de correcciones de línea y posición.

CAPÍTULO 1

ANTECEDENTES

1.1 Planteamiento del problema

En la actualidad, las industrias se ven en la necesidad de automatizar sus plantas para poder subsistir en el mundo tecnológico en el que vivimos. Esto se ve reflejado en países en los cuales a pesar de contar con mano de obra económica, integran en sus plantas industriales grandes cantidades de robots autónomos.(Faro de Vigo, 2012)

Este proyecto resuelve los problemas comunes que presentan los almacenes al no conocer las localidades de cada uno de sus productos, espacio insuficiente, desorganización y a su vez pérdida de tiempo, además de generar incontables problemas de inventario. Es por eso que hoy en día, la necesidad de automatizar es fundamental y la idea de implementar un robot autónomo encargado de almacenar y despachar productos es una gran alternativa para solucionar los problemas antes expuestos.(Palenzuela, 2016)

1.2 Justificación del problema

La idea de implementar un robot autónomo para la organización de un almacén sirve en la industria para evitar el riesgo en la salud y seguridad de las personas si los productos que se manejan atentan directamente contra su vida. El proyecto resuelve este problema, considerando que la disposición de los productos nocivos, es una actividad perjudicial que requiere planificación adecuada y un robot es el indicado para realizarlo de manera eficaz y sin peligro.

1.3 Objetivos

1.3.1 Objetivo General

Organizar la operación de un almacén inteligente en tres dimensiones identificando productos mediante códigos QR y movilizándolos a través del Robotino para el manejo de sustancias peligrosas.

1.3.2 Objetivos Específicos

- Caracterizar el Robotino para determinar sus componentes de hardware y software utilizables en el movimiento de objetos.
- Interpretar los códigos QR mediante visión artificial utilizando una tarjeta de desarrollo ARM para la identificación de las características de cada producto.
- Establecer la comunicación entre el Robotino y la tarjeta de desarrollo ARM para compartir información relacionada con el almacenamiento y despacho de productos.
- Validar el sistema en un ambiente estructurado de al menos treinta localidades de almacenamiento para analizar su desempeño.

1.4 Propuesta de solución

En la actualidad, un almacén ya no es el lugar para guardar productos de cualquier manera, ahora, es una estructura clave que provee elementos físicos y funcionales.(Palenzuela, 2016)

Es por eso que, el presente proyecto desea dar solución al problema de organización desarrollando un algoritmo que permita al Robotino almacenar productos de una forma fácil, mediante la interpretación de códigos QR, lo que permitirá sustituir la mano de obra humana por el trabajo de un robot, el cual desempeña actividades las veinticuatro horas del día y los trescientos sesenta y cinco días del año. Logrando de esta manera reducir costos de personal al igual que optimizar tiempos de almacenaje y despacho de productos.

1.5 Alcance

Este proyecto va dirigido especialmente a las industrias, para solucionar problemas ocasionados por la falta de organización, e indirectamente va dirigido a las personas que trabajan en el almacenamiento y despacho, pues realizan actividades que atentan contra su vida al tratarse de productos nocivos y de actividades repetitivas que resultan tediosas.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Robotino

El Robotino perteneciente a la marca FESTO, es un sistema para la educación en materia de robótica móvil y tecnología del automatismo.

Robotino es un robot omnidireccional, es decir, que puede desplazarse de un lugar a otro sobre una superficie plana o semi plana horizontal, gracias a las tres ruedas que posee. Además dispone de sensores analógicos y digitales, una cámara web con conexión USB y un panel de control que le permite realizar diferentes tipos de actividades sin inconvenientes. (Weber , Bellenberg, & Schwarzenberger, Manual, Robotino, FESTO, 2010). En la Figura 2.1 se puede observar la estructura del robot omnidireccional, Robotino.

Figura 2.1: Robot Robotino



Robot Omnidireccional, Fuente: (Bliesener, y otros, 2011)

2.1.1 Aplicaciones de los robots móviles

Hoy en día, los robots móviles realizan actividades comandadas por voz, son capaces de detectar y localizar objetos. Los robots móviles son un avance significativo en materia de robótica. Los sistemas móviles actuales pueden realizar actividades similares a los robots antiguos, además, gracias a sus ruedas omnidireccionales, son capaces de desplazarse de un lugar a otro en cualquier superficie horizontal. (Weber , Bellenberg, & Schwarzenberger, Manual, Robotino, FESTO, 2010).

2.1.2 Tareas en la industria

En la industria cada vez se van reemplazando sistemas de transporte autónomo, en áreas de trabajo donde existe un alto índice de peligro para el ser humano. A los sistemas que se desplazan sobre una superficie horizontal se los denomina robots móviles. Estos robots pueden realizar actividades de traslado de objetos de un lugar a otro hacia las máquinas de montaje o embalaje y recogerlas desde allí, o para realizar tareas de almacenamiento y despacho de productos. (Bliesener, y otros, 2011)

2.2 Códigos QR

Un código QR, por sus siglas en inglés, (*Quick Response Barcode*) es un sistema para el almacenamiento de información en una matriz de puntos o un código de barras bidimensional creado por la empresa japonesa Denso Wave en el año 1994. Fueron creados como una fusión de la alta densidad de información PDF417. (Martínez Rodríguez, 2014)

El código QR es una matriz cuadrada, sencillamente identificable por su patrón de búsqueda compuesto por cuadrados dentro de otros cuadrados en tres de sus esquinas. Posee un número máximo de 177 símbolos o módulos en cada lado. Son capaces de codificar 7.366 caracteres numéricos o 4.464 alfanuméricos, con la posibilidad de codificar claramente caracteres japoneses kanji y kana. Los códigos QR están elaborados para una lectura rápida por una cámara y en este momento cuentan con certificación ISO. (García & Okazaki, 2012). En la Figura 2.2 se observan los parámetros de elaboración de un código QR.

Figura 2.2: Desarrollo del Código QR



Desarrollo del código QR, Fuente: (Martínez Rodríguez, 2014)

Los códigos QR almacenan datos de manera gráfica, ya que se fundamentan en una tecnología de acaparamiento en una matriz bidimensional, almacenan un total de siete kilobytes de datos. Estos códigos pueden ser impresos en cualquier tipo de papel y por ende en cualquier impresora. Es muy importante contar con una cámara para la lectura e interpretación de códigos QR. Al igual que los demás códigos bidimensionales y que los unidimensionales, se fundamenta en patrones o diseños que consisten en módulos. El módulo es un diseño de tamaño mínimo del código. (García & Okazaki, 2012)

El identificador de patrón que poseen los códigos QR en tres de sus cuatro esquinas les permite una ubicación fácil de la posición, tamaño e inclinación. (Martínez Rodríguez, 2014).

2.2.1 Descripción de los códigos QR

La codificación QR es una matriz cuadrada con cuatro niveles de corrección de errores, tamaños normalizados y variados, que van desde 21x21 celdas, hasta 177x177 celdas y 4 tipos de información. (Martínez Rodríguez, 2014).

A continuación, en la Tabla 2.1 se muestra las especificaciones de la cantidad de caracteres por código:

Tabla 2.1: Especificaciones de la cantidad de caracteres

Numéricos	3,3 bits/caracter	7089
Alfanuméricos	5,5 bits/caracter	4296
Binarios	8 bits/caracter	2953
Kanji/Kana	13 bits/caracter	1817

Tabla de la cantidad máxima de caracteres por código según el tipo de dato. Elaborado por: Zaira Calvopiña.

Los códigos QR lucen una distribución excelente e invulnerable ante posibles daños. Aunque si el código sufre algún tipo de daño, es posible recuperar datos ya que posee la capacidad de corrección de errores.

Se establecen cuatro niveles de corrección a elegir dependiendo de la manera en la que los códigos van a ser tratados.

Teniendo en cuenta que los niveles Q y H están diseñados para ambientes de trabajo llenos de suciedad, en los cuales estos códigos puedan sufrir daños. El nivel L se utiliza para áreas de trabajo con mayor limpieza y en el cual se vayan a manejar una gran cantidad de datos. Sin embargo, el nivel de corrección más utilizado es el M. (Martínez Rodríguez, 2014)

Debido a las características presentadas anteriormente, la capacidad de almacenamiento tiene estrecha relación con el nivel de corrección de errores del código QR. Así, mientras más alto es el nivel de corrección, la capacidad de almacenamiento de información del código QR disminuye.

A continuación se presenta la Tabla 2.2 con los porcentajes de datos recuperados según el nivel de corrección:

Tabla 2.2: Niveles de corrección de códigos QR

Nivel L	7%
Nivel M	15%
Nivel Q	25%
Nivel H	30%

Porcentajes de datos recuperados según el nivel de corrección. Elaborado por: Zaira Calvopiña

Se conoce que hoy en día existe aproximadamente unas cuarenta variaciones de códigos QR, estas se diferencian por el número de módulos que contiene cada una.

La primera variación posee 21 módulos por lado, siendo esta la más reducida. Finalmente se encuentra la variación número 40, la cual contiene 177 módulos por lado.

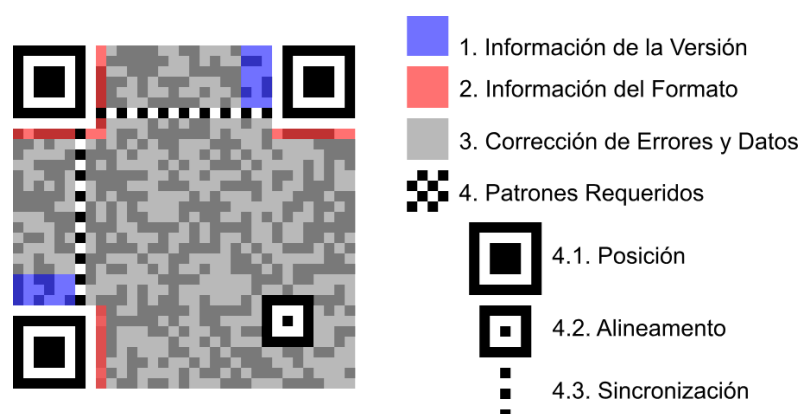
Un código QR se puede dividir hasta en 16 códigos, pues cada uno contiene en su interior otros códigos diferentes.

La característica más importante que representan estos códigos QR es que gracias a los patrones que ubicados en sus tres esquinas, se puede capturar e interpretar desde cualquier posición indicada. (Martínez Rodríguez, 2014)

Un número y una letra son los encargados de identificar a los códigos QR. El número determina la variación del código, mientras que, la letra revela el nivel de codificación aplicado.(Martínez Rodríguez, 2014)

En la Figura 2.3 se muestran los elementos que constituyen un código QR.

Figura 2.3: Elementos de un Código QR



Elementos que constituyen un código QR, modelo 2, versión 5.Fuente: (Martínez Rodríguez, 2014)

2.3 Raspberry Pi

En el año 2009 fue creada la Fundación Raspberry Pi con el único objetivo de elaborar un dispositivo accesible a cualquier persona, pero destinado especialmente a niños.

Decididos a hacer algo al respecto, la Fundación diseñó un ordenador que estimulase la enseñanza de informática básica. Upton, diseñador de sistemas integrados (SoC) de Broadcom contaba con los conocimientos y los contactos necesarios para desarrollar una solución que permita la llegada a la industria de graduados capaces de programar, con los consiguientes problemas que ello conlleva. (Volver, 2012)

2.3.1 Descripción de la placa

una gran variedad de E/S se fomenta la experimentación con el hardware. (Volver, 2012)

2.3.2 Hardware de Raspberry Pi

La tarjeta Raspberry Pi 3 Modelo B+ cuenta con una serie de elementos.

A continuación, en la Tabla 2.3 se describe cada uno de ellos y se detalla algunas características de los mismos.

Tabla 2.3: Elementos principales de la placa Raspberry Pi 3

ELEMENTO	CARACTERÍSTICA
Procesador	Cuenta con el procesador BCM 2835, de 32 bit, con una CPU ARM1176JZF-S trabajando a 700 MHz. Este modelo posee 512MB de memoria RAM.
Entrada USB	Este modelo B+, cuenta con 4 conectores USB 2.0, a diferencia del modelo anterior que solo contaba con 2.
Entrada de corriente	La conexión de la placa se lleva a cabo mediante un conector micro USB.
Conector Ethernet	Es un conector estándar RJ45. Aunque también se puede realizar conexión inalámbrica mediante un adaptador wifi conectado a un puerto USB.
Conector HDMI	Proporciona salida de video y de audio digital. La señal HDMI puede ser convertida a DVI, señal de video compuesto, o SCART mediante adaptadores externos.
Salida de audio	La placa dispone también de un conector 3.5mm mini Jack para la salida de audio.
Módulo de conexión para la cámara	1 unidad
Módulo de conexión para una pantalla LCD	1 unidad
Pines de entrada y salida	Cuenta con 40 pines, que permiten a la placa interactuar con otros dispositivos del entorno.

Elementos principales de la placa Raspberry í 3. Elaborado por: Zaira Calvopiña.

Cada uno de los pines de la placa son de gran importancia para interactuar con otros elementos, así como también para enviar y recibir cualquier tipo de información.

A continuación, en la Figura 2.5 se presenta la distribución de estos así como sus funciones.

Figura 2.5: Pines Raspberry Pi.

01	3.3V		5V	02
03	GPIO 2		5V	04
05	GPIO 3		Ground	06
07	GPIO 4		GPIO 14	08
09	Ground		GPIO 15	10
11	GPIO 17		GPIO 18	12
13	GPIO 27		Ground	14
15	GPIO 22		GPIO 23	16
17	3.3V		GPIO 24	18
19	GPIO 10		Ground	20
21	GPIO 09		GPIO 25	22
23	GPIO 11		GPIO 08	24
25	Ground		GPIO 07	26
27	ID_SD		ID_SC	28
29	GPIO 05		Ground	30
31	GPIO 06		GPIO 12	32
33	GPIO 13		Ground	34
35	GPIO 19		GPIO 16	36
37	GPIO 26		GPIO 20	38
39	Ground		GPIO 21	40

Distribución de pines de la tarjeta de desarrollo, Raspberry Pi 3. Fuente: (García Gadea, 2016)

Los pines que se encuentran en la placa cumplen diversas funciones, como son:

- Pines de alimentación: su función principal es la de suministrar voltaje. Cuenta con pines de 3,3 voltios y 5 voltios (limitados a una corriente de 50mA) y pines de tierra.
- GPIO: son 40 pines configurables, dependiendo de la programación estos pueden ser de entrada o salida. (García, D., 2016: 21)

2.3.3 Software de Raspberry Pi

Raspberry Pi funciona con varios sistemas operativos que pueden ser instalados en ella. Desde la web de Raspberry Pi, se pueden descargar estos sistemas operativos, como son: Raspbian, Windows 10, Ubuntu Mate, entre otros.

El sistema operativo Raspbian está fundamentado en Debian Wheezy de Linux. Es un sistema con aproximadamente cuatro años de desarrollo. Esto hace que sea uno de los sistemas más utilizados al ser muy avanzado y optimizado. (García Gadea, 2016)

2.4 Visión Artificial

Se define a la “Visión Artificial” como un área derivada de la “Inteligencia Artificial” que, a través de la generación de imágenes digitales y utilizando las técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información.

La visión artificial es la encargada de realizar el análisis de imágenes y está compuesta por un conjunto de procesos. Estos procesos son: captación de imágenes, memorización de la información, procesado e interpretación de los resultados.(Gobierno de España, Ministerio de Educación, 2012)

Con la visión artificial se pueden:

- Automatizar tareas de inspecciónrepetitivas realizadas por operadores.
- Realizar controles de calidad de productos.
- Realizar inspecciones de objetos sin contacto físico.
- Reducir el tiempo de ciclo en procesos automatizados.
- Realizar inspecciones en procesos donde existe diversidad de piezas.(Gobierno de España, Ministerio de Educación, 2012)

2.5 Manipulación de productos

La manipulación de productos químicos ocasiona un peligro para los seres vivos. Es de gran importancia saber cómo operarlos para evitar que dichos riesgos generen accidentes.

Las sustancias peligrosas para la salud son aquellas que pueden causar contusiones ingresando al cuerpo por una o varias vías a la vez. Una única sustancia puede generar contusiones en varias zonas del cuerpo humano. (Universidad Autónoma de Occidente, 2011)

2.5.1 Producto Químico

Un producto químico, es un conjunto de compuestos sintéticos destinados a cumplir una función. Generalmente el que cumple la función principal es un solo componente,

llamado componente activo. Los compuestos restantes o excipientes, son para llevar a las condiciones óptimas al componente activo y darle mejor aspecto. (Universidad Autónoma de Occidente, 2011)

Al investigar en la página de la Agencia para Sustancias Tóxicas y el Registro de Enfermedades se obtuvo información de las sustancias nocivas al ser humano las cuales causan daño en la salud y por las cuales es preferible utilizar un robot para remplazar la mano de obra humana.

En la Tabla 2.4 se realiza la descripción de cada sustancia y el daño que estas ocasionan en la salud del ser humano.

Tabla 2.4: Descripción de sustancias nocivas para el ser humano

Producto	Peligro para el ser humano
Acetona	Irritación de la nariz, garganta, pulmones y ojos; dolores de cabeza, mareo, confusión, aceleración del pulso, náusea, vómito, pérdida del conocimiento y posiblemente coma.
Acroleína	Lagrimo de los ojos, ardor de nariz y garganta, reducción del ritmo respiratorio, daña los pulmones y puede producir la muerte.
Aluminio	Desarrolla problemas respiratorios tales como tos o alteraciones que se visualizan en radiografías del tórax, provoca enfermedades de los huesos o del cerebro.
Amianto	Afecta principalmente a los pulmones y a la membrana que los envuelve pudiendo causar una enfermedad grave que eventualmente puede producir incapacidad y la muerte.
Amoníaco	Irrita la piel, ojos, garganta, pulmones y puede producir tos y quemaduras, además produce daño del pulmón y hasta la muerte.
Anilina	Daña a la hemoglobina que transporta el oxígeno en la sangre. También provoca mareo, dolores de cabeza, latido irregular del corazón, convulsiones, coma y la muerte.
Antimonio	Irrita los ojos y pulmones, puede causar trastornos del corazón y de los pulmones, dolor de estómago, diarrea, vómitos y úlceras estomacales.
Arsénico	Produce dolor de garganta e irritación de los pulmones, náusea y vómitos, disminución del número de glóbulos rojos y blancos, ritmo cardíaco anormal, fragilidad capilar y una sensación de hormigueo en las manos y los pies.
Bario	Altera el ritmo del corazón y produce parálisis y posiblemente la muerte.
Benceno	Produce somnolencia, mareo, aceleración del ritmo cardíaco, dolor de cabeza, temblores, confusión y pérdida del conocimiento. Además produce alteraciones en la médula de los huesos y produce anemia.

Tabla 2.4: Descripción de sustancias nocivas para el ser humano

Berilio	Produce la enfermedad aguda de berilio, esta enfermedad puede hacerlo sentirse débil y cansado, también puede causar anorexia, pérdida de peso, y en casos avanzados, producir enfermedades del corazón. La exposición prolongada aumenta la posibilidad de desarrollar cáncer del pulmón.
Boro	Afecta al estómago, intestinos, hígado, riñón y el cerebro y eventualmente puede causar la muerte.
Cadmio	Daña los riñones y posiblemente causa una enfermedad renal, daña los pulmones y produce fragilidad de los huesos.
Cianuro	Causa dificultad para respirar, dolores del pecho, vómitos, alteraciones en la sangre, dolores de cabeza y dilatación de la glándula tiroides, daña el cerebro y el corazón y puede producir coma y la muerte.
Cloro	Irritación de la nariz, garganta y ojos, tos y alteraciones del ritmo respiratorio y daño de los pulmones.
Cromo	Irritación del revestimiento interno de la nariz, úlceras nasales, secreción nasal y problemas respiratorios tales como asma, tos, falta de aliento o respiración jadeada.
Endrina	Produce lesiones graves al sistema nervioso y la muerte. Tragar grandes cantidades puede producir convulsiones y la muerte en unos pocos minutos u horas.
Etión	Produce náusea, sudor, diarrea, pérdida del control de la vejiga, visión borrosa, temblores musculares y dificultad para respirar. La intoxicación grave puede producir coma, incapacidad para respirar y la muerte.
Fenol	Irritación de las vías respiratorias, dolor de cabeza y ardor en los ojos. El contacto de la piel ha producido quemaduras de la piel, daño del hígado, orina de color oscuro, latido irregular del corazón y en algunos casos, la muerte.
Formaldehído	Irritación a la piel, ojos, nariz y garganta. Beber grandes cantidades de formaldehído puede causar profundo dolor, vómitos, coma, y posiblemente la muerte.
Fosgeno	Irritación de los ojos y garganta haciéndolo toser y jadear, dilatación de los pulmones, dificultando así la respiración. Los niveles altos pueden producir daño grave en los pulmones que puede causar la muerte.
Gasolina	Produce efectos graves al sistema nervioso incluyendo coma e inhabilidad para respirar y efectos de menor gravedad incluyen mareo y dolores de cabeza.
Mercurio	Irritabilidad, timidez, temblores, alteraciones a la vista o la audición y problemas de la memoria. Además de lesiones al pulmón, náusea, vómitos, diarrea, aumento de la presión sanguínea o del pulso, salpullidos e irritación a los ojos.

Tabla 2.4: Descripción de sustancias nocivas para el ser humano

Naftalina	Destruye una porción de sus glóbulos rojos. Algunos de los síntomas de la anemia hemolítica son fatiga, falta de apetito, agitación y palidez, puede causar náusea, vómitos, diarrea, sangre en la orina y una coloración amarillenta de la piel
Plomo	Afecta a casi todos los órganos y sistemas en el cuerpo. También puede producir debilidad en los dedos, muñecas o tobillos, y puede causar anemia. La exposición a niveles altos puede dañar seriamente el cerebro, los riñones y causar la muerte.
Plutonio	El efecto principal de la exposición es el cáncer. Los tipos de cáncer que ocurren más comúnmente son cáncer del pulmón, el hígado y los huesos.
Radón	Desarrolla cáncer del pulmón.
Tolueno	Produce cansancio, confusión, debilidad, sensación de embriaguez, pérdida de la memoria, náusea y pérdida del apetito. Además puede causar daño permanente del cerebro.
Uranio	Inhalar o ingerir compuestos de uranio ha dañado el riñón.
Xileno	Produce dolores de cabeza, mareo, confusión y alteraciones del equilibrio, causar irritación de la piel, ojos, nariz y garganta, problemas pulmonares, retardo del tiempo de reacción a estímulos. La exposición a niveles muy altos puede causar pérdida del conocimiento y la muerte.

Tabla de sustancias nocivas y peligros que causan en el ser humano. Fuente: (Agencia para Sustancias Tóxicas y el Registro de Enfermedades, 2016)

2.5.2 Vías de entrada del contaminante al cuerpo

Las sustancias químicas son absorbidas por el organismo humano por las vías:

- **Vía respiratoria:** Es una de las entradas principales de entrada de la sustancia tóxica al organismo en las diferentes actividades laborales. Por esta vía las sustancias ingresan forma de partículas y gases.
- **Vía dérmica:** Los compuestos tóxicos ingresan por la piel y se adhieren al organismo, produciendo efectos que abarcan desde irritación local hasta sensibilización de la persona e inclusive la muerte.
- **Vía digestiva:** El ingreso de sustancias tóxicas por esta vía, se genera por falta de higiene al comer o manipular alimentos en áreas de trabajo, sin embargo esta vía no es muy habitual aparentemente para la contaminación.

- **Vía parenteral:** Las sustancias tóxicas no solo ingresan a través de la piel sana, sino también por contusiones en la piel expuestas al entorno laboral, lo cual aumenta el riesgo de daño al organismo (Universidad Autónoma de Occidente, 2011).

La Figura 2.6 muestra las vías de entrada del contaminante químico al cuerpo.

Figura 2.6: Vías de entrada del contaminante al cuerpo



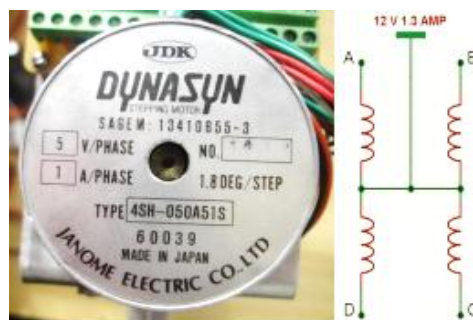
Vías de entrada de contaminantes al cuerpo. Fuente: (Universidad Autónoma de Occidente, 2011)

2.6 Elementos Electrónicos

- **Motor a pasos DYNASYN**

El motor a pasos DYNASYN es un motor Unipolar, posee 6 cables de conexión: negro y blanco para alimentación, un par de cables color rojo y un par de cables color verde representativos a las cuatro bobinas que posee. En la Figura 2.7 se muestra el motor a pasos DYNASYN que es utilizado para el control del eje vertical en "z". (Cuji Cuji, 2013)

Figura 2.7: Motor a pasos DYNASYN



Motor a pasos DYNASYN, para el control del eje móvil. Elaborado por: (Cuji Cuji, 2013)

Es importante mencionar que se ha elegido este motor por sus propiedades de fuerza y torque esenciales para el correcto movimiento del eje en “z”.

En la Tabla 2.5 que se presenta a continuación se muestra algunas características importantes del motor a pasos.

Tabla 2.5: Características del motor a pasos DYNASYN

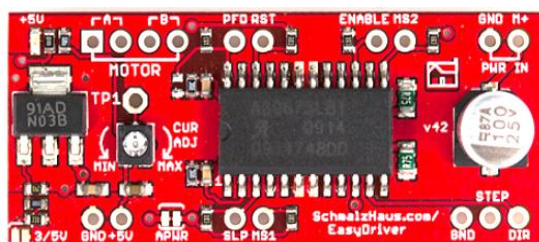
MOTOR A PASOS DYNASYN	
Marca:	DYNASYN
Tipo:	4SH-050A1S
Voltaje por fase:	5V
Amperios por fase:	1A
Número de pasos:	1.8 DEG/STEP
Número de bobinas:	4

Tabla de características importantes del motor a pasos unipolar DYNASYN. Elaborado por: Zaira Calvopiña.

- **Easy Driver**

El EasyDriver v4.4 que se muestra en la Figura 2.8, es un pequeño dispositivo utilizado para el control de motores a pasos, fundamentado en el chip controlador Allegro A3967, soporta una corriente de hasta aproximadamente 750mA por fase de un motor paso a paso. Se implanta el modo de microstepping de 8 pasos por defecto. (Schmalz, 2015)

Figura 2.8: EasyDriver v4.4



Módulo EasyDriver v4.4 para controlar el motor a pasos DYNASYN. Fuente: (Schmalz, 2015)

El módulo Easy Driver soporta una corriente máxima variable de aproximadamente 150mA/fase a 750mA/fase. Puede tomar un voltaje máximo del motor de aproximadamente 30V e incluye una regulación de 5V incorporada. (Schmalz, 2015)

- **Módulo LM2596 Convertidor de Voltaje DC-DC Buck**

Es un dispositivo regulador de voltaje. Soporta una corriente aproximada de 3A. Maneja una carga con excelente regulación de línea y bajo voltaje de rizado. Este regulador posee dos pines de entrada y dos pines de salida, los cuales son regulables, gracias al potenciómetro lineal que posee. El módulo mostrado en la Figura 2.9 reduce al mínimo el uso de componentes externos como baterías de lipo. (ELECTRONILAB)

Figura 2.9: Módulo LM2596 Convertidor de Voltaje DC-DC Buck



Módulo LM2596 Convertidor de Voltaje DC-DC Buck. Fuente: (ELECTRONILAB)

En la Tabla 2.6 que se presenta a continuación se muestra algunas características importantes del módulo LM2596.

Tabla 2.6: Características Módulo LM2596

Módulo LM2596 Convertidor de Voltaje DC-DC Buck	
Voltaje de entrada:	4.5-40V
Voltaje de Salida:	1.5-35V (Ajustable)
Corriente de Salida:	Máxima 3A
Frecuencia de Switching:	150KHz

Tabla de características importantes del Módulo LM2596 Convertidor de Voltaje DC-DC Buck. Fuente: (ELECTRONILAB)

- **Sensor de campo magnético**

Este sensor permite detectar de manera sencilla, rápida y precisa la presencia de campos magnéticos o electromagnéticos en cualquier sitio en el cual este se localice.

Este sensor de efecto Hall mostrado en la Figura 2.10 mide campos magnéticos. Es un sensor digital, es por eso que cuando se detecta la presencia de un campo magnético se enviará una señal en alto, es decir, un 1 lógico, caso contrario se enviará un 0 lógico. (Mil y un partes, 2014)

Figura 2.10: Sensor de Campo Magnético



Sensor de campo magnético de efecto Hall. Fuente: (Mil y un partes, 2014)

En la Tabla 2.7 que se presenta a continuación se muestra algunas características importantes del Sensor de campo magnético de efecto Hall.

Tabla 2.7: Características importantes del Sensor de campo magnético de efecto Hall

Sensor de Campo Magnético	
Voltaje de funcionamiento:	3.3V-5V
Interruptor digital salida:	(0/1)

Tabla de características importantes del Sensor de campo magnético de efecto Hall. Fuente: (Mil y un partes, 2014)

- **Servomotor**

Es un mecanismo semejante a un motor de corriente directa, mediante programación, este puede localizarse en cualquier ubicación dentro de un rango de trabajo e inclusive mantenerse en esa posición hasta que se le ordene algún cambio.

Los servomotores se manejan mediante modulación por ancho de pulsos (PWM). El servomotor consta de tres cables. El terminal de señal es de color amarillo, gracias a este se controla el pulso de entrada al servomotor. El cable de color negro es el correspondiente al neutro y el cable de color rojo es el correspondiente a la alimentación (5V). (tecmikro). En la Figura 2.11 se muestra un servomotor Hitec HS-311.

Figura 2.11: Servomotor Hitec HS-311



Servomotor para el control de la pinza sujetadora. Fuente: (tecmikro)

En la Tabla 2.8 que se presenta a continuación se muestra algunas características importantes del servomotor Hitec HS-311.

Tabla 2.8: Características importantes del Servomotor Hitec HS-311

Servomotor Hitec HS-311	
Torque máximo:	51 onzas-pulgada
Velocidad máxima:	0,15 s/60grados
Desplazamiento angular:	90 grados
Posición neutral:	1.5 ms
Voltaje de operación:	4,8-6.0V
Dirección:	Multi-direccional
Engranajes:	Nylon
Peso:	43 gramos

Tabla de características importantes del Servomotor Hitec Hs-311. Fuente: (tecmikro)

- **Cámara Web Logitech C170**

Logitech® Webcam C170. Posee configuración Plug and Play con resolución VGA en Logitech Vid™ HD. Esta cámara maneja una resolución de hasta 5 megapíxeles, para obtener y enviar información. Consta de un micrófono integrado con reducción de ruido. Graba vídeos nítidos y con buenos colores en resolución XVGA (1024 x 768). La Figura 2.12 que se muestra a continuación, posee un soporte de ajuste para cualquier superficie en la que se desee ubicarla. (Logitech, 2011)

Figura 2.12: Cámara Web Logitech C170



Cámara Web Logitech C170. Fuente: (Logitech, 2011)

En la Tabla 2.6 que se presenta a continuación se muestran los datos técnicos de la cámara Logitech C170.

Tabla 2.6: Especificaciones técnicas de la cámara Logitech

Especificaciones técnicas – Cámara Logitech C170	
Tecnología	Logitech Fluid Crystal™*
Fotos	Hasta 5 megapíxeles (mejora por software)
Micrófono	Integrado con reducción de ruido
Certificación	USB 2.0 de alta velocidad
Software	Logitech Vid™ HD
Captura de video	Hasta 1024 x 728
Videoconferencias	640x 480 pixeles

Especificaciones técnicas de la cámara Logitech. Fuente: (Logitech, 2011)

CAPÍTULO 3

DISEÑO, DESARROLLO E IMPLEMENTACIÓN

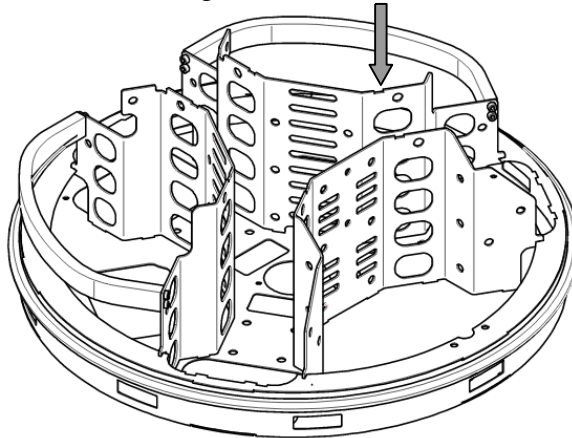
3.1 Caracterización Robot Robotino

El Robotinopropio de la marca alemana FESTO, es el robot indicado para la automatización de procesos industriales en los que se requiere un robot inteligente totalmente autónomo. Gracias a cada uno de los elementos que lo componen, es capaz de realizar cualquier tipo de actividad para la cual este programado, siempre y cuando se tenga en cuenta las limitaciones del mismo. Robotino está formado por un número elevado de elementos. A continuación, se presentan cada uno de sus principales componentes con breves características de los mismos. En este apartado se iniciará con los elementos que son indispensables para el control y funcionamiento del robot. (Weber, Bellenberg, & Schwarzenberger, 2010)

3.1.1 Chasis

El chasis está fabricado de acero inoxidable y con soldadura de láser. En esta plataforma es donde están montados todos los elementos físicos que componen Robotino. En la Figura 3.1 se presenta el chasis de acero inoxidable de Robotino.(Weber, Bellenberg, & Schwarzenberger, 2010)

Figura 3.1: Chasis



Chasis de acero inoxidable de Robotino. Fuente: (Weber, Bellenberg, & Schwarzenberger, 2010)

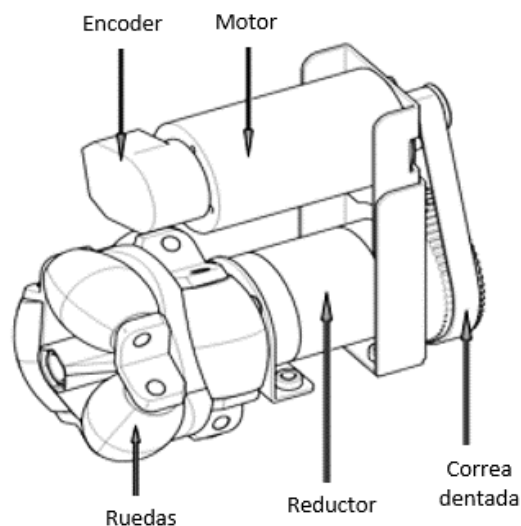
3.1.2 Baterías

Robotino posee un par de baterías recargables montadas en el chasis del robot. Estas están fabricadas de plomo/ácido de 12 voltios, las cuales alimentan al robot con una tensión de 24 voltios de corriente continua. Cada una de estas baterías posee un fusible automotriz de 5 amperios. (Weber, Bellenberg, & Schwarzenberger, 2010)

3.1.3 Unidad de accionamiento (Omnidrive)

La unidad de accionamiento presente en la Figura 3.2, es la responsable del movimiento del robot. Consta de tres módulos de accionamiento omnidireccionales independientes que están montados en el chasis, cada uno ubicado a 120° entre sí. (Weber, Bellenberg, & Schwarzenberger, 2010)

Figura 3.2: Unidad de accionamiento



Elementos que forman la unidad de accionamiento de Robotino. Fuente: (Weber, Bellenberg, & Schwarzenberger, 2010)

En este apartado se presentan datos técnicos de cada componente de la unidad de accionamiento.

- Motor DC. Robotino posee tres motores DC, los cuales pueden controlarse con un regulador PID. (Weber, Bellenberg, & Schwarzenberger, 2010)

Algunas de las especificaciones se muestran en la Tabla 3.1.

Tabla 3.1: Valores Técnicos del Motor a pasos DC.

Motor DC	
Tensión Nominal	24 VDC
Velocidad Nominal	3600 RPM
Velocidad sin carga	4200 RPM
Intensidad nominal	0,9 A
Intensidad de arranque	4 A
Intensidad sin carga	0,17 A

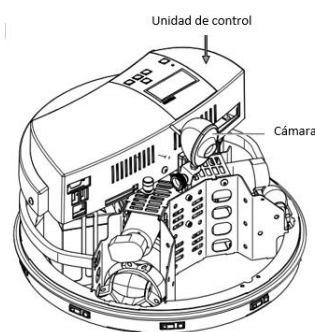
Tabla de valores técnicos de las características del motor DC. Elaborado por: Zaira Calvopiña

- Reductor Planetario. Permite que el robot se desplace a bajas velocidades y que sea muy preciso en sus movimientos, posee una relación de reducción de 16:1.
- Encoder Incremental. Determina la posición del robot, gracias a los valores que emite este encoder.
- Ruedas omnidireccionales. Permiten el movimiento del robot sobre su propio eje, así como también el desplazamiento del mismo. (Weber, Bellenberg, & Schwarzenberger, 2010)

3.1.4 Unidad de control

Esta unidad es el cerebro principal del robot, se encarga de enviar diferentes instrucciones a cada uno de los elementos conectados a ella. En la Figura 3.3 se observa la unidad de control del robot.(Weber, Bellenberg, & Schwarzenberger, 2010)

Figura 3.3: Unidad de Control



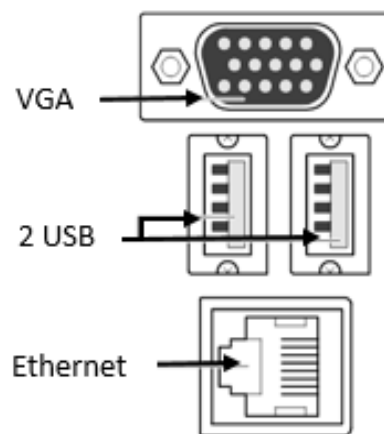
Unidad de control de Robotino. Fuente: (Weber, Bellenberg, & Schwarzenberger, 2010)

Esta unidad está equipada de elementos internos que permiten su correcto funcionamiento:

- Procesador. PC104 con 300 MHz
- Sistema Operativo: Linux con Kernel en tiempo real
- SDRAM: 128 M
- Tarjeta Compact Flash: 1 GB
- Puerto de acceso: LAN inalámbrico.

La unidad de control consta de tres tipos de interfaces: Ethernet, USB y VGA. Las cuales permiten la conexión de un teclado, un ratón y un monitor, convirtiendo de esta forma a la tarjeta en un mini ordenador capaz de realizar cualquier tipo de proyecto electrónico. En la Figura 3.4 se muestran las interfaces de la unidad de control.(Weber, Bellenberg, & Schwarzenberger, 2010)

Figura 3.4: Interfaces de la Unidad de Control



Interfaces de la Unidad de control de Robotino. Fuente: (Weber, Bellenberg, & Schwarzenberger, 2010)

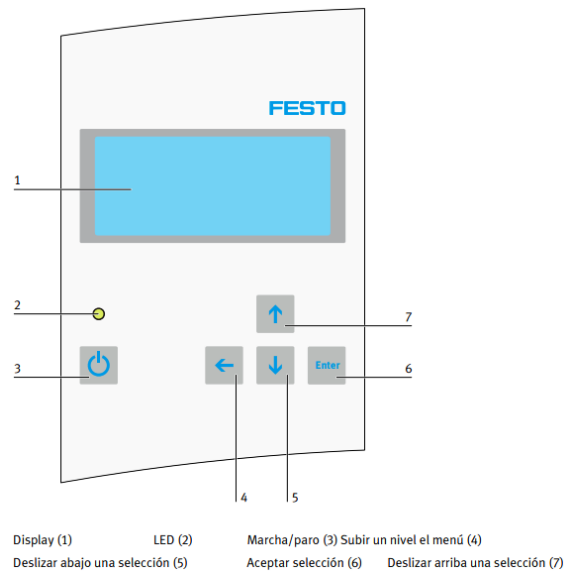
3.1.5 Teclado y Display

Se encuentran ubicados en la parte superior de la unidad de control, el teclado posee cinco teclas desde las cuales se puede seleccionar entre diferentes programas preestablecidos en el robot como por ejemplo trayectoria circular, avance, cuadrilátero, explorar y seguidor de línea.

Así como también permite seleccionar entre opciones de: nivel de batería, estado de carga, idiomas y configuración de red cuya información se la puede observar en el display. (Weber, Bellenberg, & Schwarzenberger, 2010)

En la Figura 3.5 se puede visualizar el teclado y display de Robotino.

Figura 3.5: Teclado y display



Teclado y display del robot indica el nombre del robot, la ip, y la versión. Fuente: (Weber, Bellenberg, & Schwarzenberger, 2010)

3.1.6 Sensores

Robotino posee una serie de sensores que le permiten ser un robot autónomo e inteligente. Gracias a la ayuda de estos, puede desplazarse por cualquier trayectoria previamente programada, evadiendo obstáculos, siguiendo una línea, e inclusive girando sobre su propio eje.

Los sensores presentes en el robot son:

- **Sensor de distancia por infrarrojos.** Suman un total de nueve sensores en el robot, estos se encuentran ubicados en el chasis y distribuidos a 40° entre sí. Estos sensores pueden medir distancias de entre 4 y 30 centímetros.
- **Sensor anticoliisión.** Se encuentra ubicado alrededor del chasis del robot, cumple la función de impedir que se interrumpa la ejecución de un programa

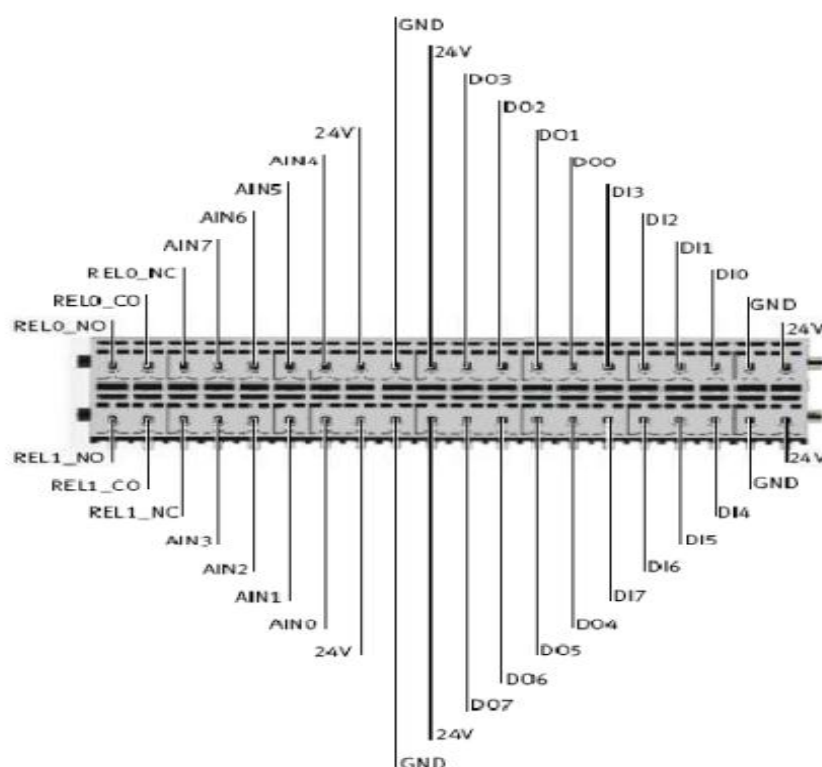
al efectuarse alguna colisión, además si se produce una colisión con algún objeto este sensor detiene el movimiento de Robotino.

- **Sensor de proximidad inductivo analógico.** Se encuentra ubicado en la parte inferior del robot, su función es la de detectar objetos metálicos en la superficie que se desplace el robot.
- **Sensor de reflexión directa.** El seguimiento de una ruta también puede ser implementado con los dos sensores de reflexión directa (de luz difusa) incluidos.
- **Cámara.** Se encuentra ubicada en la parte frontal de Robotino, se conecta a la unidad de control a través del puerto USB. (Weber, Bellenberg, & Schwarzenberger, 2010)

3.1.7 Interfaces

En la Figura 3.6 se muestran las interfaces de E/S que permiten conectar sensores y actuadores adicionales al robot.

Figura 3.6: Interfaces E/S



Interfaces de la unidad de accionamiento. Fuente: (Weber, Bellenberg, & Schwarzenberger, 2010)

La barra de interfaces posee diversos puertos:

- 8 entradas analógicas (0 a 10 V) (AIN0 hasta AIN7)
- 8 entradas digitales (DI0 hasta DI7)
- 8 salidas digitales (DO0 hasta DO7)
- 3 relés para actuadores (REL0 y REL1). Estos contactos pueden utilizarse como contacto normalmente abierto (NA=NO), contacto normalmente cerrado (NC) a (CO= colusorio) conmutados. (Weber, Bellenberg, & Schwarzenberger, 2010)

3.2 Configuración de la placa Raspberry Pi 3

A continuación, se explica el proceso de iniciación del sistema operativo de la placa Raspberry Pi y de las librerías a utilizar, además de la instalación de programas tanto para la detección de códigos QR como para la creación de la interfaz web.

3.2.1 Inicialización del Sistema Operativo

Para la inicialización del Sistema Operativo, se descargó el mismo desde la página: <https://www.raspberrypi.org/downloads/raspbian/> original de Raspberry sin ningún inconveniente ya que este sistema es software libre.

Se utilizó una tarjeta micro SD, para este proyecto se ha elegido una tarjeta micro SD Samsung de 32 Gb, clase 10, 4K FullHD con una velocidad de lectura de 95 Mbps y velocidad de escritura de 20 Mbps.

Se formateó la tarjeta micro SD utilizando el software SDFormatter como FAT32 y finalmente utilizando el software Win32DiskImager, se procedió a instalar el sistema operativo en la tarjeta micro SD.

3.2.2 Iniciación de Apache y PHP.

Para la creación de la interfaz web se instaló dos software necesarios para el correcto funcionamiento de la misma. Estos softwares son: Apache y Php.

- El servidor Apache HTTP, también llamado **Apache**, es utilizado para la creación de páginas y servicios web. Es un servidor multiplataforma,

gratuito, muy sólido y que sobresale por su seguridad y rendimiento. (Ibrugor, 2014)

A continuación se presenta la Figura 3.7 en la cual se puede observar una secuencia de pasos a seguir para la instalación de Apache.

Figura 3.7: Proceso de instalación de Apache

Descarga	Descarga la última versión desde http://httpd.apache.org/download.cgi
Extraer	<pre>\$ gzip -d httpd-NN.tar.gz \$ tar xvf httpd-NN.tar \$ cd httpd-NN</pre>
Configura	<pre>\$./configure --prefix=PREFIX</pre>
Compila	<pre>\$ make</pre>
Instala	<pre>\$ make install</pre>
Personalizalo	<pre>\$ vi PREFIX/conf/httpd.conf</pre>
Prueba	<pre>\$ PREFIX/bin/apachectl -k start</pre>

Pasos a seguir para el proceso de iniciación de Apache. Fuente: (Apache HTTP Server Project, 2018)

- PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy conocido y utilizado para el desarrollo web y que puede ser encajado en HTML.

A continuación se presenta un listado de códigos, que se siguen paso a paso para la instalación de PHP dentro de la Raspberry Pi desde consola.

- `apt-get install php5 php5-mysql php5-gd libapache2-mod-php5`
- `/etc/init.d/apache2 restart`
- `vi /var/www/testphp.php`
- `<?php`
// Muestra toda la información, por defecto INFO_ALL
`phpinfo();`
`?>`
- `http://127.0.0.1/testphp.php`

Se prueba la validez de PHP ingresando la ip del servidor. (Medina, 2013)

3.3 Desarrollo del Sistema de Lectura e Interpretación de Códigos QR

Para dar inicio al desarrollo del sistema de lectura e interpretación se procedió a generar los códigos QR debido a que fueron estos la clave fundamental de arranque para el sistema general de almacenamiento y despacho de productos. Estos códigos son de tipo texto y llevan el nombre de sustancias nocivas.

3.3.1 Generación de Códigos QR

Cabe destacar que el almacén consta de treinta localidades para cada uno de los productos, es decir, se generaron treinta códigos QR. Estos códigos llevar el nombre de una sustancia nociva. Lo que significa que se generaron treinta códigos QR con treinta tipos de productos nocivos para el ser humano.

Como es de conocimiento general, en internet existe un sin número de páginas para generar códigos QR con diferentes características, dependiendo de las necesidades de los usuarios.

Para realizar este proyecto se ha elegido <http://www.codigos-qr.com/generador-de-codigos-qr/>, ya que es uno de los generadores más completos y en el cual se puede elegir inclusive el idioma. Ya que a más de ser un Generador de Códigos QR es un Generador de Código de Barras y a su vez un Generador Datamatrix.

Este generador permite crear una gran gama de códigos, esto gracias a que se puede elegir entre opciones como: direcciones URL, SMS, TELÉFONO, EMAIL, TEXTO, VCARD, GEOLOC, EVENTO O WIFI como se muestra en la Figura 3.8 para este proyecto se ha elegido la opción de TEXTO.

Figura 3.8: Interfaz del Generador de Códigos QR.

Generador de Códigos QR





URL  SMS  TELÉFONO  EMAIL  **TEXTO ** VCARD  GEOLOC  EVENTO  WIFI 

Código QR para Texto

Introduce un texto para generar un Código QR o QR Code. Este código una vez leído desde un móvil, permitirá al usuario leer el texto

Texto:

Interfaz Software Generador de Códigos QR, opción TEXTO. Fuente: (Códigos QR, 2009)

Una vez elegida la opción TEXTO, se procedió a escribir el nombre de la sustancia nociva, se ha optado por escribir el nombre de cada sustancia sin tildes para no cometer errores posteriores en la programación.



En la Figura 3.9 se observa que el generador nos permite además seleccionar el tipo de tamaño del código así como la redundancia del mismo.

Figura 3.9: Opciones para generar el código QR


Texto:

ARSENICO

152 caracteres disponibles

Tamaño  Redundancia 

Mediano Alta


 GENERAR CÓDIGO QR

Selección del tamaño y redundancia del código QR. Fuente: (Códigos QR, 2009)

Con lo que respecta a Tamaño, se puede elegir entre: muy pequeño, pequeño, mediano, grande y muy grande. Y con lo que respecta a redundancia, se refiere a la legibilidad, es decir, que este factor permite al código QR ser leído inclusive si ha sufrido algún tipo de deterioro. Se puede elegir entre: baja, media, alta y muy alta. En este proyecto se ha elegido un tamaño mediano y una redundancia alta. A continuación, en la Figura 3.10 se muestra un ejemplo de la creación del código QR del arsénico.

Figura 3.10: Código QR – ARSENICO



Descarga este código QR pulsando el botón derecho del
ratón o usa siguiente permalink 

`<img src="http://www.codigos-qr.com/qr/php/qr_img.php?d=Al`

Código QR del Arsénico. Elaborado por: Zaira Calvopiña

En la parte inferior de la Figura 3.10, se puede observar un permalink, el cual sirve para descargar el código QR, o a su vez se lo puede descargar pulsando el botón derecho del ratón como se indica.

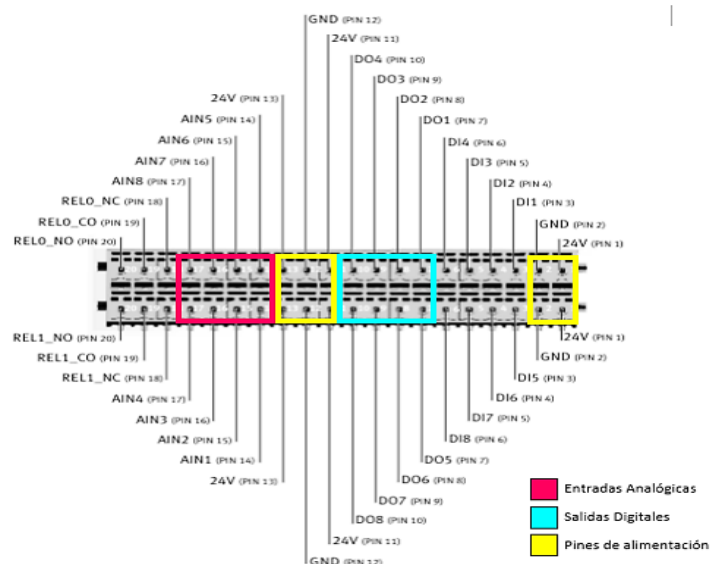
3.4 Diseño del Sistema Electrónico

En este apartado se detalla el desarrollo y la implementación del sistema de control electrónico para Robotino. Es importante mencionar que el robot posee un eje móvil en “z” y una pinza sujetadora en su estructura, por lo que este proyecto se enfoca en el control electrónico de los mismos.

Para realizar el control se utilizó una tarjeta de adquisición de datos Raspberry Pi 3. Esta tarjeta es el cerebro principal de todo el proyecto, ya que en ésta se desarrollan los algoritmos de control de cada una de las partes que conforman el mecanismo móvil del robot así como también el algoritmo para la lectura e interpretación de códigos QR.

Es importante mencionar que para la implementación del sistema de almacenamiento y despacho de productos en Robotino, se realizó la caracterización de las borneras del robot, tomando en cuenta que se necesitan entradas analógicas, salidas digitales y pines de alimentación, en la Figura 3.11 se presentan las interfaces E/S del Robotino.

Figura 3.11: Interfaces E/S utilizadas de Robotino



Interfaces E/S de Robotino. Elaborado por: Zaira Calvopiña.

3.4.1 Circuito de control para eje móvil en “z”

Para realizar el control del movimiento del eje móvil en “z” se utilizó un motor a pasos DYNASYN, un Easy Driver, un Módulo LM2596 y un sensor de campo magnético cuyas especificaciones técnicas se encuentran detalladas en el capítulo 2.

Se estableció un diagrama de bloques para realizar el control del movimiento del eje móvil. Como se muestra en la Figura 3.12, se alimenta el circuito con 24V provenientes de Robotino, haciendo uso del módulo LM2596, es decir, ingresa al módulo un voltaje de 24V a la entrada y se obtiene un voltaje de 5V a la salida, lo que permitió alimentar el Easy Driver y el sensor de campo magnético.

Una vez establecida la alimentación, se procedió a conectar las bobinas A y B del motor al Easy Driver y adicionalmente los pines GPIO9 (STEP), GPIO 10 (DIR), y GND hacia la Raspberry, estos pines son los encargados del control de dirección y arranque del motor a pasos. También se conectó, el pin GPIO 7 para la señal del sensor de campo magnético y los pines GPIO 5 y GPIO 11 correspondientes a los finales de carrera.

Finalmente se conectaron tres salidas digitales de las borneras del robot mediante los opto acopladores hacia los pines GPIO 18, GPIO 23 y GPIO 24 de la Raspberry para posteriormente usar estas señales en la programación del algoritmo de control final. En la Figura 3.12 se muestra el diagrama de bloques para el control del eje móvil.

Figura 3.12: Diagrama de Bloques para el control del eje móvil en “z”

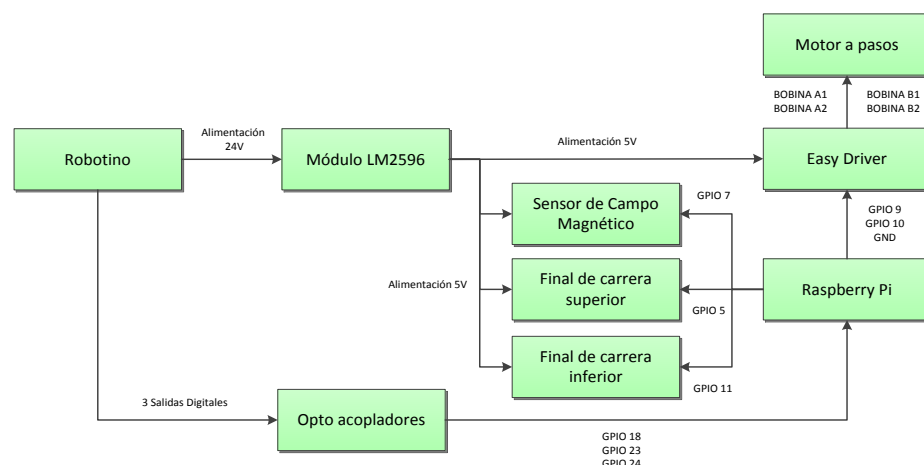


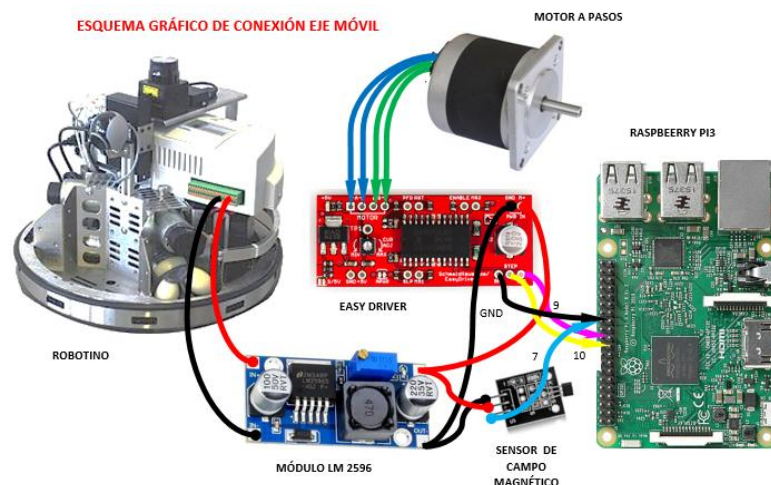
Diagrama de Bloques para el control del eje móvil en “z”. Elaborado por: Zaira Calvopiña.

Es importante mencionar que el algoritmo de control del eje móvil en “z” se lo realizó en Python 2.3 y se lo puede revisar detalladamente en el ANEXO I.

- **Diagrama de conexión gráfica para eje móvil en “z”**

A continuación en la Figura 3.13 se presenta el esquema gráfico de conexión de los elementos que conforman el mecanismo del eje del robot.

Figura 3.13: Esquema gráfico de conexión gráfica del eje móvil en “z”



Esquema gráfico de conexión para eje móvil. Elaborado por: Zaira Calvopiña.

3.4.2 Circuito de control pinza sujetadora

Para realizar el control del movimiento de la pinza sujetadora se utilizó un servomotor y un Módulo LM2596 cuyas especificaciones técnicas se encuentran detalladas en el capítulo 2.

Se estableció un diagrama de bloques para realizar el control de la pinza sujetadora, se alimenta el circuito con 24V provenientes de las baterías de Robotino, pero haciendo uso del módulo convertidor de voltaje, es decir, ingresa al módulo un voltaje de 24V a la entrada y obteniendo un voltaje de 5V a la salida, lo que permitió alimentar el servomotor. Una vez establecida la alimentación, se procedió a conectar el pinGPIO 12 de la señal del servomotor.

Finalmente se conectaron dos salidas digitales de las borneras del robot mediante los opto acopladores hacia los pines GPIO 8 y GPIO 16 de la Raspberry Pi para

posteriormente usar estas señales en la programación del algoritmo de control final de todo el proyecto.

Es importante mencionar que el algoritmo de control de la pinza sujetadora se lo realizó en Python 2.3 y se lo puede revisar detalladamente en el ANEXO II.

En la Figura 3.14 se muestra el diagrama de bloques de las conexiones para el control de la pinza sujetadora.

Figura 3.14: Diagrama de Bloques para las conexiones de la pinza sujetadora

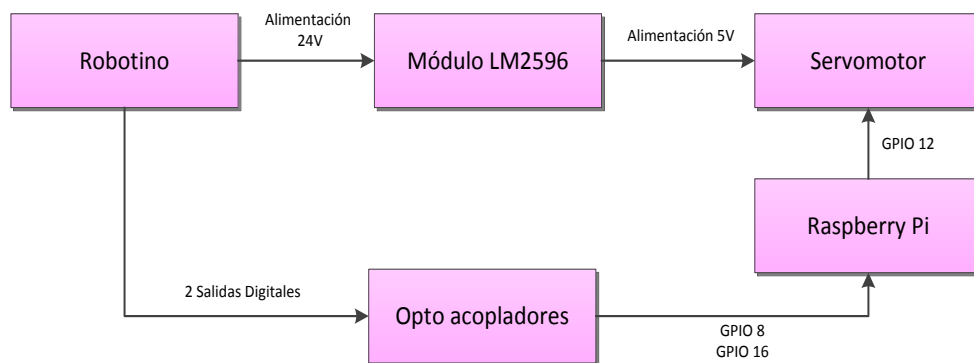
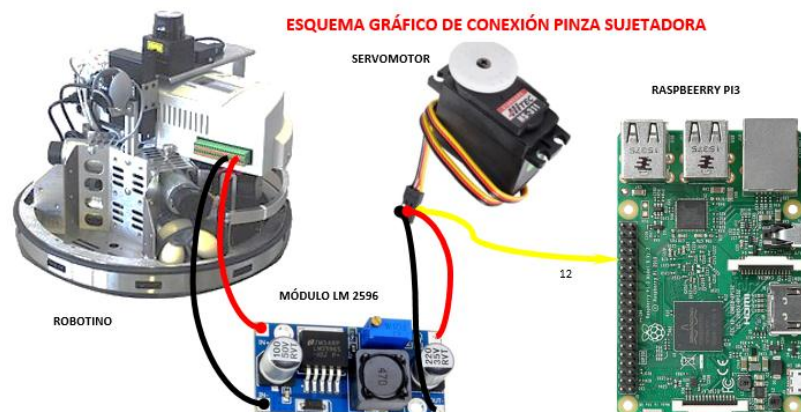


Diagrama de Bloques para las conexiones de la pinza sujetadora. Elaborado por: Zaira Calvopiña.

- **Diagrama de conexión gráfica para pinza sujetadora**

A continuación en la Figura 3.15 se presenta un esquema gráfico de conexión de los elementos que conforman el mecanismo de la pinza sujetadora.

Figura 3.15: Esquema gráfico de conexión gráfica para pinza sujetadora



Esquema gráfico de conexión para pinza sujetadora. Elaborado por: Zaira Calvopiña.

3.4.3 Circuito de control para lectura e interpretación de códigos QR

Para realizar el control de la lectura e interpretación de códigos QR se utilizó una cámara Logitech conectada directamente en la Raspberry Pi.

- **Diagrama de Bloques para la Lectura e Interpretación de Códigos QR**

En este apartado se presenta la Figura 3.16 en la que se observa el diagrama de bloques de las conexiones entre la cámara, Raspberry Pi y Robotino con sus respectivas señales.

Figura 3.16: Diagrama de Bloques para la Lectura e Interpretación de Códigos QR

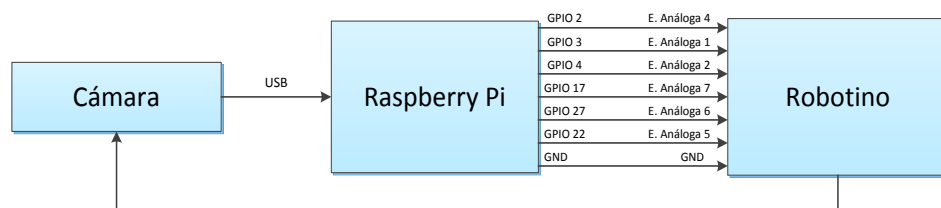


Diagrama de Bloques para la Lectura e Interpretación de Códigos QR. Elaborado por: Zaira Calvopiña.

Como se puede observar en el diagrama, la cámara está conectada mediante cable USB a la Raspberry Pi y es la encargada de leer los códigos QR y enviar información del mismo hacia la tarjeta. Los pines GPIO 2, 3, 4, 17, 27 y 22 de la Raspberry Pi se conectan al Robotino mediante sus entradas analógicas 4, 1, 2, 7, 6 y 5 respectivamente.

El algoritmo de control para lectura e interpretación permite capturar el código QR mostrado a la cámara y envía a la tarjeta Raspberry Pi la información del nombre del producto y por ende la localidad en la que éste se debe ubicar.

Es importante mencionar que el algoritmo de lectura e interpretación de códigos QR se lo realizó en Python 2.3 y se lo puede revisar detalladamente en el ANEXO III.

Para la interpretación de los códigos QR se ha establecido trabajar con una cadena de seis bits, ya que al ser treinta productos, se necesitan un total de sesenta combinaciones de seis bits, treinta combinaciones para el almacenamiento y treinta combinaciones para el despacho.

Por lo que al disponer de seis bits tendremos $2^6 = 64$, donde 6 es el número de bits y 64 el número de combinaciones de bits posibles. Se decidió trabajar con las combinaciones de los números impares para el almacenamiento y con las combinaciones de números pares para el despacho. Es decir un producto tendrá una combinación de seis bits para el almacenamiento y otra combinación de 6 bits para el despacho. De esta manera cuando se requiera almacenar o despachar un productor no existirá ninguna confusión. En la Tabla 3.2 se muestran las combinaciones para cada uno de los productos.

Tabla 3.2: Cadena de bits por producto para el almacenamiento y despacho

Número	Producto	Almacenamiento	Despacho
1	Acetona	000001	000010
2	Acroleína	000011	000100
3	Aluminio	000101	000110
4	Amianto	000111	001000
5	Amoníaco	001001	001010
6	Anilina	001011	001100
7	Antimonio	001101	001110
8	Arsénico	001111	010000
9	Bario	010001	010010
10	Benceno	010011	010100
11	Berilio	010101	010110
12	Boro	010111	011000
13	Cadmio	011001	011010
14	Cianuro	011011	011100
15	Cloro	011101	011110
16	Cromo	011111	100000
17	Endrina	100001	100010
18	Etión	100011	100100
19	Fenol	100101	100110
20	Formaldehído	100111	101000
21	Fosgeno	101001	101010
22	Gasolina	101001	101100
23	Mercurio	101101	101110
24	Naftalina	101111	110000
25	Plomo	110001	110010
26	Plutonio	110011	110100
27	Radón	110101	110110
28	Tolueno	110111	111000
29	Uranio	111001	111010
30	Xileno	111011	111100

Cadena de bits por producto para el almacenamiento y despacho. Elaborado por: Zaira Calvopiña.

En el ANEXO IV se presenta un esquema de los pines utilizados en la Raspberry Pi tanto para el circuito de control del eje móvil en “z”, como para el circuito de control de la pinza sujetadora, así como también para el circuito de control de la lectura e interpretación de códigos QR.

3.4.4 Diagrama esquemático de la placa de opto acopladores

Las salidas digitales del Robotino emiten señales de 24V, estos voltajes son acondicionados mediante una placa de seis opto acopladores y reducido de 24V a 3.3V. Los opto acopladores son los encargados de separar tierras, por un lado la tierra del Robotino y por otro la tierra de la tarjeta Raspberry Pi. Para elaborar la placa antes mencionada se utiliza: opto acopladores 4N25, resistencias de 1K Ω , 1.2K Ω y 390 Ω . A continuación en la Figura 3.17 se presenta el diagrama esquemático de la placa de opto acopladores realizada en Proteus 8.6.

Figura 3.17: Diagrama esquemático de la placa de opto acopladores

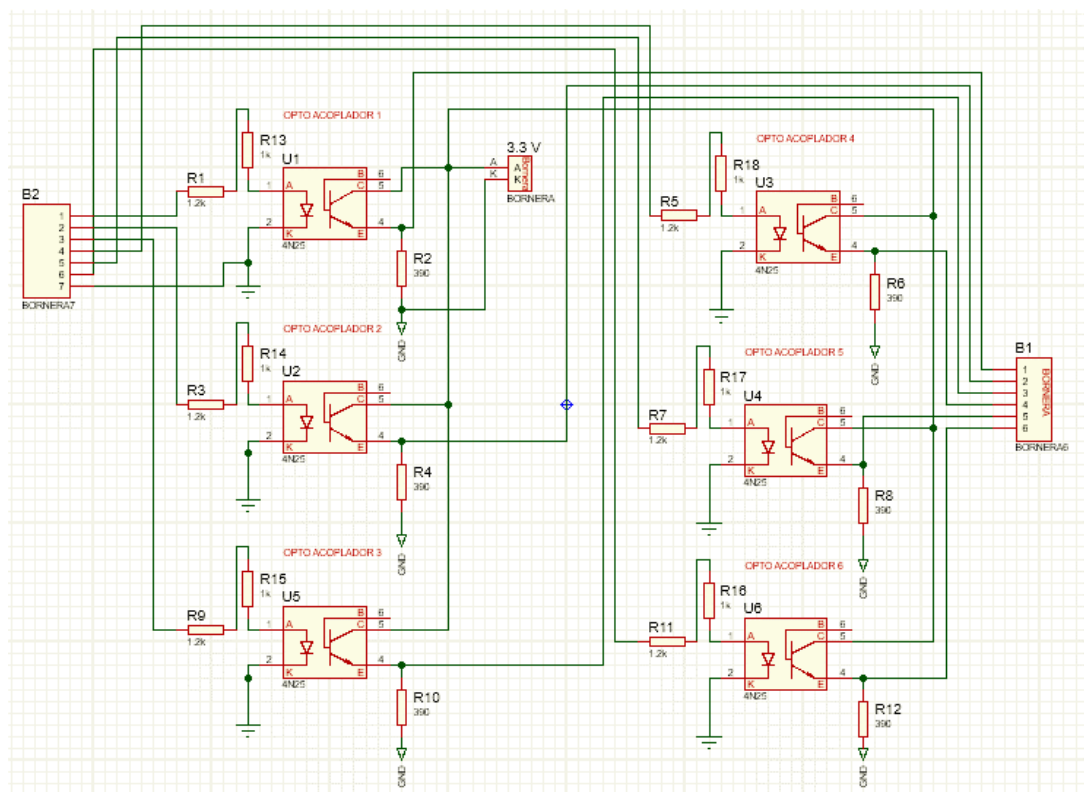
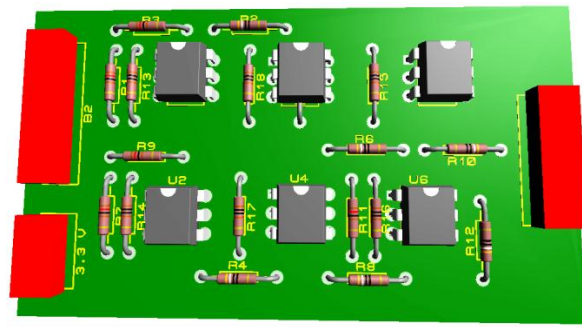


Diagrama esquemático de la placa de opto acopladores. Elaborado por: Zaira Calvopiña.

Utilizando PCB Layout se generaron las pistas de la placa de opto acopladores para su posterior impresión en una baquelita, obteniendo el resultado que se muestra en el Anexo V.

A su vez, luego de organizar las pistas se generó el diseño 3D de la placa, el cual sirve para visualizar la posición final de los elementos, así como también la forma de soldarlos a la misma. La Figura 3.18 que se presenta a continuación muestra el diseño 3D de la placa.

Figura 3.18: Diseño 3D de la placa de opto acopladores

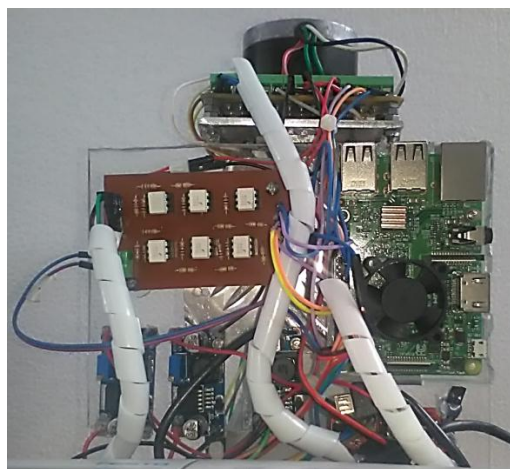


Diseño 3D de la placa de opto acopladores, en la cual se visualizan los elementos utilizados.

Elaborado por: Zaira Calvopiña.

Una vez finalizada la placa de opto acopladores, se procedió a ensamblar todos los circuitos y módulos respectivos, como se muestra en la Figura 3.19, con el objetivo de obtener un robot compacto que se desplace de manera autónoma.

Figura 3.19: Conexión total de los elementos de control al robot



Implementación total de los elementos del diseño electrónico. Elaborado por: Zaira Calvopiña.

3.4.5 Diagrama de Flujo del Sistema de Almacenamiento de productos.

Para la realización del diagrama de flujo que rige todo el sistema de almacenamiento y despacho de producto, se acopló el sistema de lectura e interpretación de códigos QR y la estructura conformada por el eje móvil y la pinza sujetadora. En la Figura 3.20 se presenta el diagrama de flujo del sistema de almacenamiento de productos.

Figura 3.20: Diagrama de flujo del proceso de almacenamiento de productos.

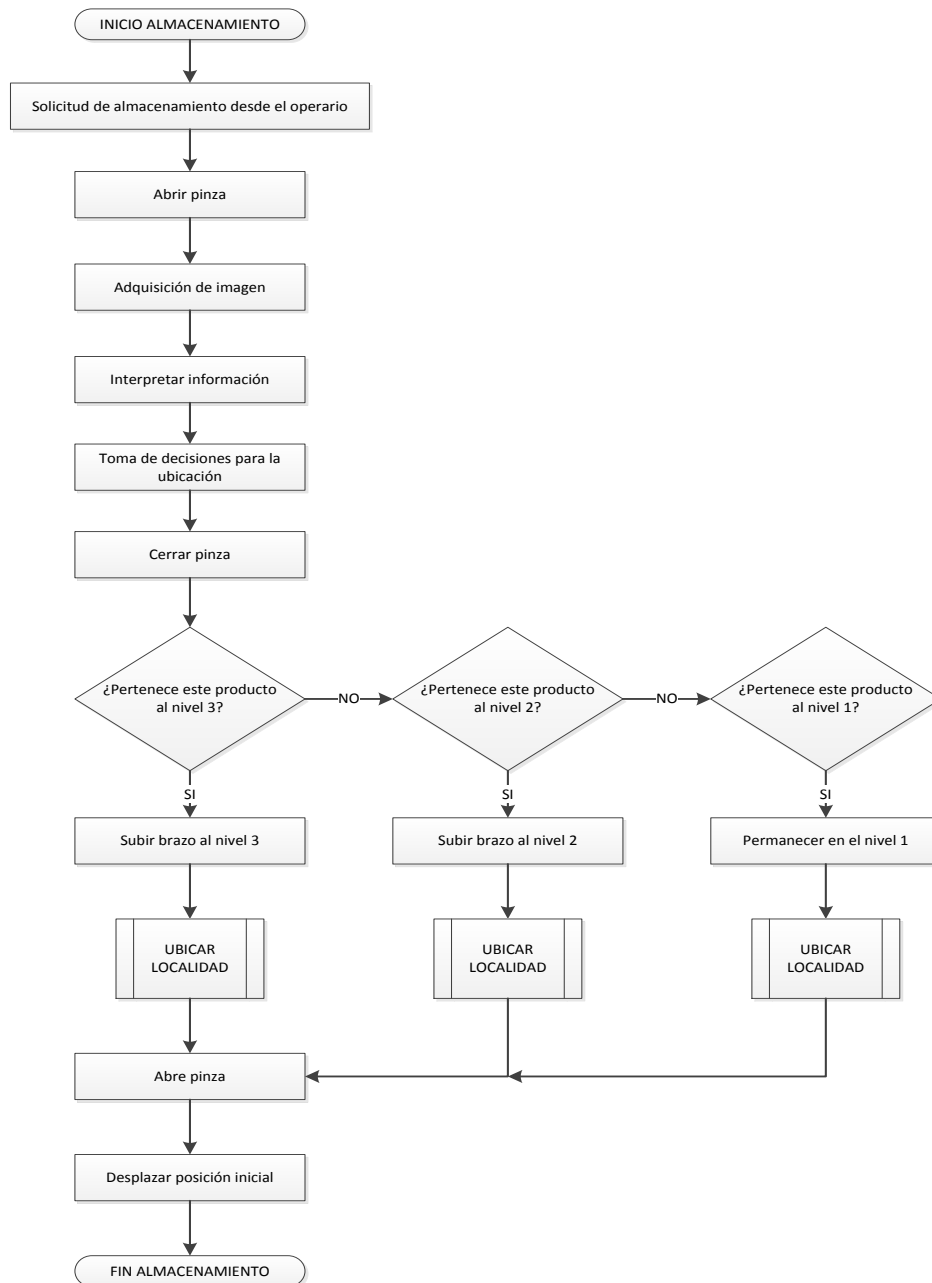


Diagrama de flujo del proceso de almacenamiento de productos. Elaborado por: Zaira Calvopiña.

3.4.6 Diagrama de Flujo del Sistema de Despacho de productos.

Si bien el proceso de despacho es lo contrario del proceso de almacenamiento de productos, este presenta una pequeña diferencia en lo que respecta a la apertura y cierre de la pinza sujetadora. A continuación, en la Figura 3.21 se presenta el diagrama de flujo del proceso de despacho de productos.

Figura 3.21: Diagrama de flujo del proceso de despacho de productos.

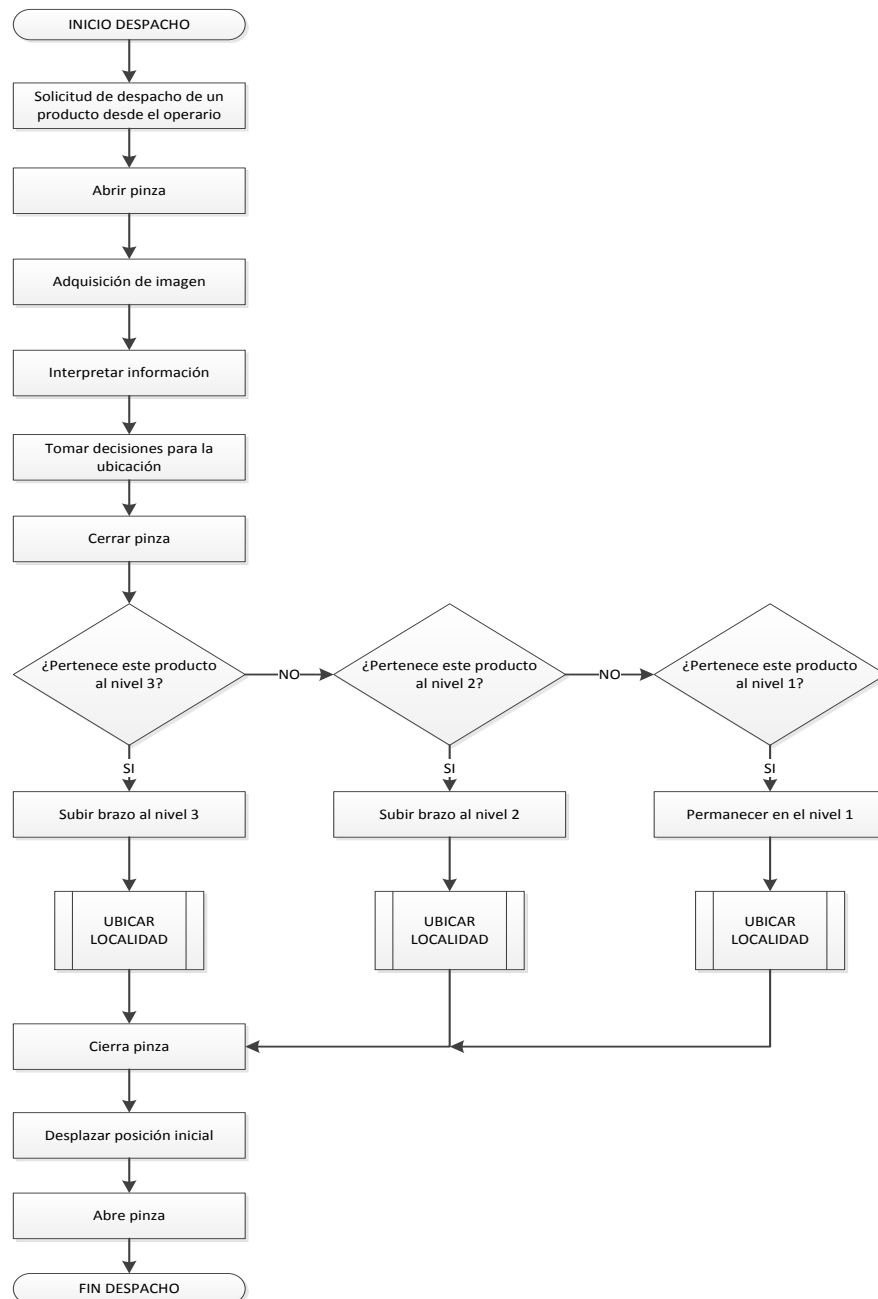
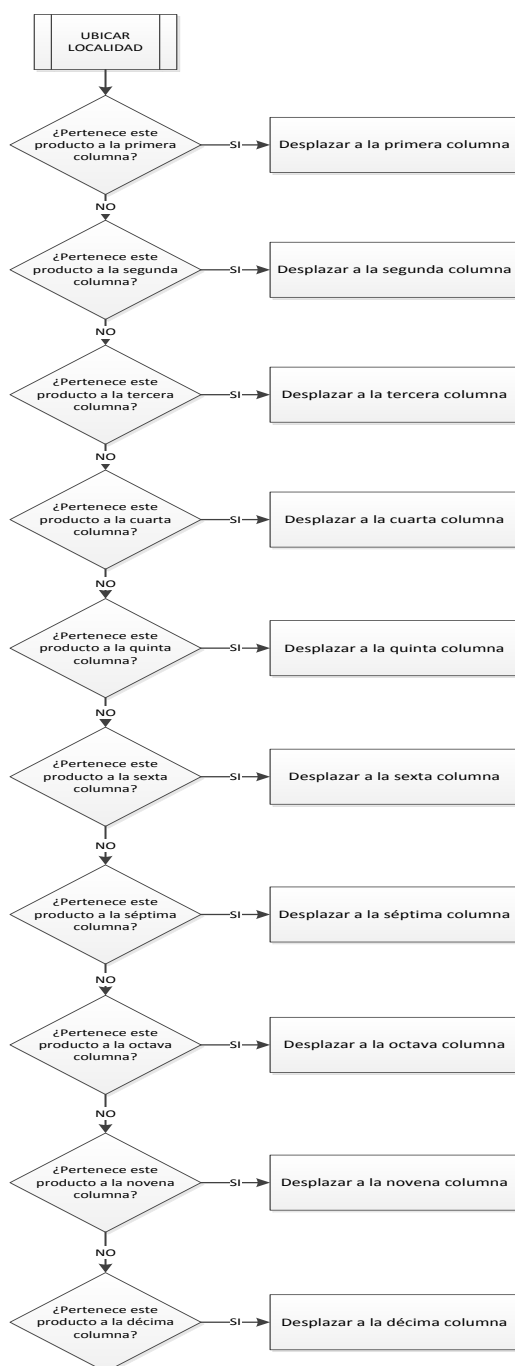


Diagrama de flujo del proceso de despacho de productos. Elaborado por: Zaira Calvopiña.

Debido a que se trata de un almacén tridimensional estructurado de tres filas y diez columnas el diagrama de flujo se vuelve un tanto complejo, por lo que se ha realizado un subproceso, el cual se presenta a continuación en la Figura 3.22.

Figura 3.22: Subproceso del diagrama de flujo del almacenamiento y despacho de productos.



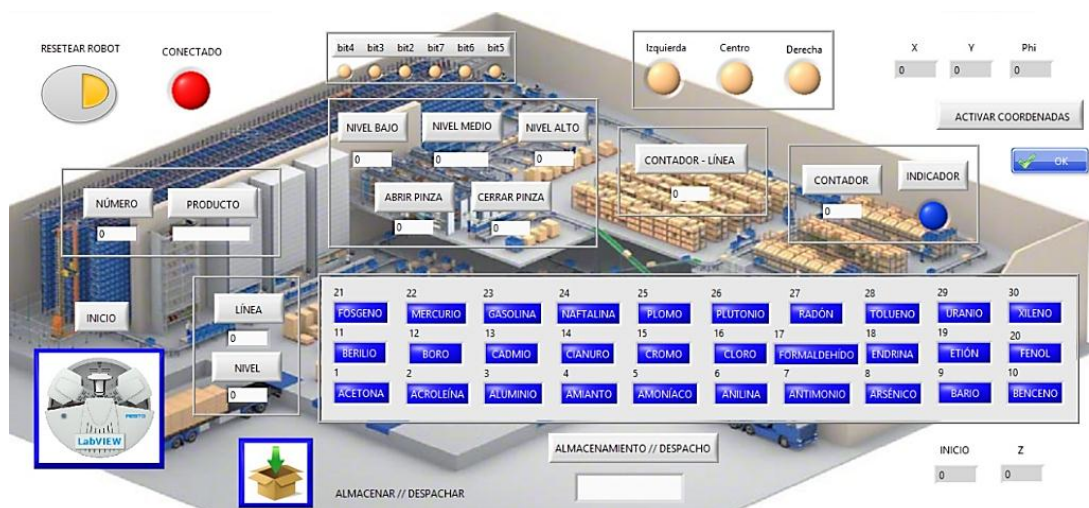
Subproceso del diagrama de flujo de almacenamiento y despacho de productos. Elaborado por: Zaira Calvopiña.

Cabe destacar que el algoritmo de control general desarrollado en Labview 2015 se presenta en el ANEXO VI.

3.4.7 Interfaz en Labview para la puesta en marcha del sistema

Para la puesta en marcha del sistema de control general del robot, se creó una interfaz gráfica en Labview, la cual se observa en la Figura 3.23.

Figura 3.23: Interfaz gráfica para el control general de Robotino



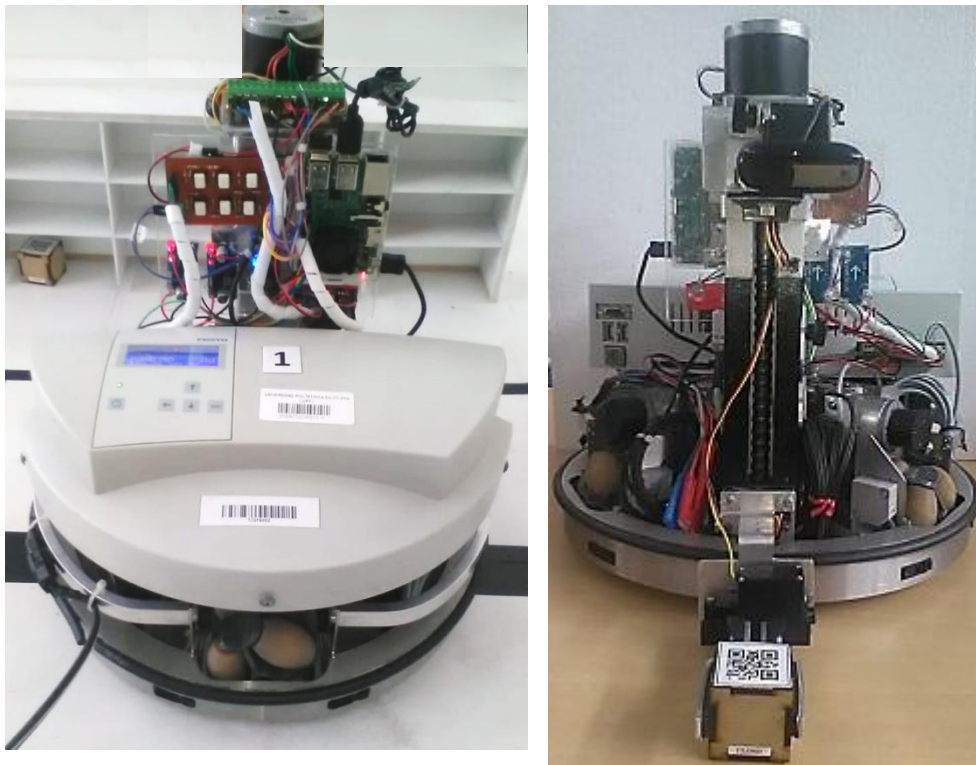
Interfaz gráfica para puesta en marcha del sistema de control de Robotino. Elaborado por: Zaira Calvopiña.

Esta interfaz consta con un espacio que representa el almacén con sus treinta localidades, un botón con la imagen del robot que permite poner en marcha el sistema, un botón con la imagen de una caja para seleccionar la opción de almacenamiento o despacho, así como también un indicador para visualizar el proceso elegido, seis leds que representan la combinación de seis bits para diferenciar cada producto, un botón de reseteo total, un led indicador de conectividad, un par de indicadores de número y nombre del producto, al igual que un par de indicadores de número de línea y nivel, tres led indicadores del funcionamiento de los sensores de reflexión directa, tres indicadores para visualizar las coordenadas del robot, un led indicador para el conteo de línea y finalmente indicadores de estado de nivel del eje móvil y de la pinza sujetadora.

3.4.8 Resultados de la Implementación Final del Sistema en el Robotino

Una vez realizadas las conexiones de cada uno de los circuitos del sistema de almacenamiento y despacho de productos, se organizan en una placa de acrílico y ésta se monta sobre la estructura del robot, convirtiéndolo en un robot compacto listo para la puesta en marcha. En la Figura 3.24 se muestra el robot compacto con la placa de acrílico.

Figura 3.24: Robot compacto con los circuitos electrónicos



Robot compacto con los circuitos electrónicos. Elaborado por: Zaira Calvopiña.

Con el robot listo para la puesta en marcha, se elaboró el almacén en el cual el robot va a realizar su trabajo. El almacén consta de treinta localidades distribuidas uniformemente en tres filas y diez columnas. Además de un ambiente estructurado de dos metros por un metro y medio en el cual el robot realiza su desplazamiento sobre las líneas guías dibujadas en éste.

En la Figura 3.25 se muestra el ambiente estructurado y el almacén con sus treinta localidades.

Figura 3.25: Ambiente estructurado y almacén



Ambiente estructurado y almacén con treinta localidades. Elaborado por: Zaira Calvopiña.

Para validar el proceso de almacenamiento y despacho, se elaboró treinta cajitas de madera que representan los productos nocivos para el ser humano, cada cajita posee el código QR de su respectivo producto. En la Figura 3.26 se muestra la cajita antes mencionada.

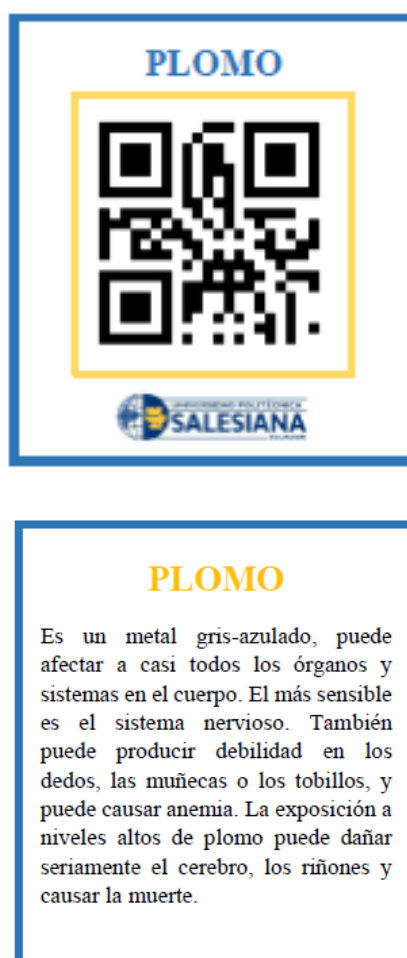
Figura 3.26: Caja de madera con código QR



Caja de madera que representa el producto nocivo para el ser humano. Elaborado por: Zaira Calvopiña

Finalmente para la validación del proceso de despacho, se elaboró treinta tarjetas con los códigos QR y las características de cada uno de los productos. Estas tarjetas se mostrarán frente a la cámara del robot para que éste sepa que producto tiene que despachar. En la Figura 3.27 se muestra la tarjeta para el despacho del PLOMO. Como se puede observar la tarjeta posee dos caras, en la cara frontal se exhibe el código QR y el nombre del producto, mientras que en la cara posterior se exhibe la característica del producto así como también los riesgos que causa a la salud del ser humano de acuerdo a los niveles la exposición al producto.

Figura 3.27: Tarjeta para el despacho de productos



Tarjeta para el despacho con su respectivo código QR y características del producto. Elaborado por:
Zaira Calvopiña

CAPÍTULO 4

PRUEBAS Y RESULTADOS

En este capítulo se presentan las pruebas realizadas para verificar el correcto funcionamiento del sistema de almacenamiento y despacho de productos utilizando el robot omnidireccional Robotino y visión artificial.

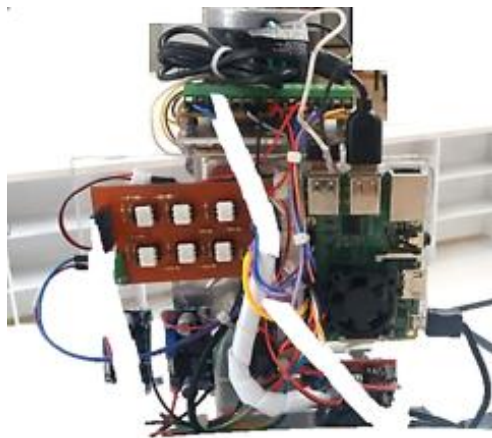
4.1 Pruebas de Hardware

Las pruebas de hardware se iniciaron principalmente con la calibración de los tres sensores de reflexión directa, estos sensores se activan cuando censan línea negra emitiendo un 1 lógico, y cuando captan el color blanco emiten un 0 lógico que desactiva el sensor.

Además se prueba que la Raspberry Pi envíe correctamente los bits de cada una de sus salidas activadas. Antes de encender el sistema, se prueba que los tres módulos convertidores de voltaje estén emitiendo correctamente un voltaje de 5 voltios, los cuales alimentan la Raspberry Pi, el motor a pasos, el servomotor y el sensor magnético respectivamente.

Con lo que respecta a la cámara es importante ubicarla en la posición correcta para que tenga un buen rango de visión y enfoque para leer los códigos QR. En la Figura 4.1 se observa el robot con el resultado final de todas las conexiones realizadas y de todos los elementos bien posicionados.

Figura 4.1: Conexiones electrónicas realizadas en el robot



Conexiones y elementos incorporados en el robot. Elaborado por: Zaira Calvopiña.

4.2 Pruebas de Software

Con lo que respecta a las pruebas de software, es muy importante mencionar que primero se cargó los algoritmos de control en la Raspberry Pi. Estos algoritmos son: motor.py, raspi.py y pinza.py.

Una vez cargados los programas mencionados, se procedió a probar la funcionalidad de los mismos utilizando LabVIEW 2015 y la respectiva librería de Robotino. Se ha elegido este software ya que es muy versátil y se puede realizar programación tanto gráfica como línea a línea manejando la estructura Fórmula Node.

4.2.1 Prueba de lectura e interpretación de códigos QR

La primera prueba que se realizó es la lectura e interpretación de códigos QR, en la cual se obtiene excelentes resultados. El robot lee correctamente los treinta códigos y no se observa problema alguno.

Tabla 4.1: Resultados obtenidos en la lectura de códigos QR

Productos leídos	Porcentaje de acierto
30	100%

Tabla de resultados obtenidos en la lectura de códigos QR. Elaborado por: Zaira Calvopiña.

4.2.2 Prueba de ascenso y descenso del eje móvil en “z”

Utilizando tres botones en el panel frontal de LabVIEW, se observa que al presionar el primer botón el eje móvil sube sin ningún problema hasta activar el final de carrera correspondiente; lo mismo ocurre al presionar el segundo botón, el eje sube a la mitad hasta activar el sensor magnético y de igual forma al presionar el tercer botón el eje móvil baja hasta activar el final de carrera.

4.2.3 Prueba de apertura y cierre de pinza sujetadora

Utilizando un par de botones en el panel frontal al igual que en el caso anterior, al presionar un botón la pinza se abre correctamente y si se presiona un segundo botón la pinza se cierra.

4.3 Pruebas en pista

Una vez realizadas las pruebas básicas de hardware y software se probó el funcionamiento de todo el sistema en general.

Inicialmente se elaboró una pista, la cual será de color blanco con líneas guías de color negro como se muestra en la Figura 4.2.

Figura 4.2: Pista para desplazamiento del robot



Pista para desplazamiento de robot Robotino. Elaborado por: Zaira Calvopiña.

En la Figura 4.2 se observa que la línea negra señalada es más gruesa que las otras, esto debido a que los sensores están ubicados formando un triángulo y estas al ser gruesas permiten la lectura correcta de los sensores.

Entre el sensor central y los sensores de izquierda y derecha se ha programado la corrección de línea horizontal que impide que el robot salga de su trayectoria. Lo que justifica el grosor de la línea mencionada es el espacio que existe entre estos para activar cada uno de los sensores.

Se realizaron las pruebas del funcionamiento final del robot, como se observa en la Figura 4.3, el almacén consta de treinta localidades, distribuidas uniformemente en tres filas y diez columnas como se mencionó en el capítulo 3, en el apartado 3.4.8.

Se implementó de la misma forma en el ambiente estructurado once líneas verticales de color negro, las cuales son las encargadas de guiar al robot a cada una de las localidades del almacén, siendo la línea cero el inicio general del proceso.

Figura 4.3: Almacén y Robotino listos para la puesta en marcha del sistema



Almacén y Robotino listo para la puesta en marcha del sistema de almacenamiento y despacho de productos. Elaborado por: Zaira Calvopiña.

Una vez lista la pista se validó el sistema de almacenamiento y despacho de productos realizando una serie de pruebas en un ambiente estructurado de treinta localidades y se analizó su funcionamiento.

4.3.1 Corrección de Línea y Corrección de Posición

Inicialmente se almacenó uno a uno los productos para observar el comportamiento del robot al desplazarse por cada una de las líneas. Se obtuvo datos con respecto al número de correcciones de línea que realiza el robot en la línea gruesa horizontal, al igual que el número de correcciones de posición que realiza al llegar casi a su punto de inicio. Obteniendo los siguientes resultados.

Cabe recalcar que en la Tabla 4.2 que se muestra a continuación existen iniciales las cuales significan: CL = corrección de línea, CP = corrección de posición, I = ida, R = regreso, OC = objetivo cumplido.

Tabla 4.2: Resultados de la corrección de línea y corrección de posición por producto

Nº	PRODUCTO	LÍNEA	PRUEBA 1				PRUEBA 2				PRUEBA 3			
			CL		CP	OC	CL		CP	OC	CL		CP	OC
			I	R			I	R			I	R		
1	Acetona	1	0	0	0	Si	0	0	0	Si	0	0	0	Si
2	Berilio	1	0	0	0	Si	0	0	0	Si	0	0	0	Si
3	Fosgeno	1	0	0	0	Si	0	0	0	Si	0	0	0	Si
4	Acroleína	2	0	0	0	Si	0	0	0	Si	0	0	0	Si
5	Boro	2	0	0	0	Si	0	0	0	Si	0	0	0	Si
6	Mercurio	2	0	0	0	Si	0	0	0	Si	0	0	0	Si
7	Aluminio	3	0	0	2	Si	0	0	1	Si	0	0	2	Si
8	Cadmio	3	0	0	2	Si	0	0	0	Si	0	0	1	Si
9	Gasolina	3	0	0	2	Si	0	2	3	Si	0	0	0	Si
10	Amianto	4	0	0	0	Si	0	0	1	Si	0	0	1	Si
11	Cianuro	4	0	0	2	Si	0	1	3	Si	0	0	1	Si
12	Naftalina	4	0	0	0	Si	0	0	0	Si	0	0	1	Si
13	Amoniaco	5	0	1	1	Si	0	0	1	Si	0	2	2	Si
14	Cromo	5	0	0	1	Si	0	0	2	Si	0	1	2	Si
15	Plomo	5	0	1	2	Si	0	1	2	Si	0	0	1	Si
16	Anilina	6	0	0	1	Si	0	0	2	Si	0	1	1	Si
17	Cloro	6	0	0	1	Si	0	1	2	Si	0	1	1	Si
18	Plutonio	6	0	1	2	Si	0	0	1	Si	0	1	2	Si
19	Antimonio	7	0	1	1	Si	0	1	2	Si	0	1	1	Si
20	Formaldehído	7	0	1	2	Si	0	1	1	Si	0	0	2	Si
21	Radón	7	0	1	1	Si	0	1	2	Si	0	1	2	Si
22	Arsénico	8	2	9	4	Si	2	7	5	Si	3	8	4	Si
23	Endrina	8	0	8	5	No	4	6	4	Si	3	9	5	No
24	Tolueno	8	1	9	5	Si	3	6	5	No	2	7	4	Si
25	Bario	9	1	3	4	Si	1	2	4	Si	1	3	3	No
26	Etién	9	1	4	4	No	1	2	3	No	1	2	2	Si
27	Uranio	9	2	3	2	No	1	3	4	No	1	3	2	No
28	Benceno	10	5	10	4	Si	4	8	6	No	5	6	8	Si
29	Fenol	10	4	10	6	No	6	10	5	Si	7	10	7	No
30	Xileno	10	4	10	5	No	7	9	8	No	8	9	6	No

Resultados de la corrección de línea y corrección de posición que realiza el robot al desplazarse por la pista para almacenar y despachar cada uno de los productos. Elaborado por: Zaira Calvopiña.

Al analizar los datos obtenidos se observó que mientras mayor es la distancia que recorre Robotino mayor es el número de correcciones de línea y de posición que realiza el robot.

En las dos primeras líneas que recorre el robot no existe ninguna corrección, en este punto se puede decir que el robot realiza su proceso fácilmente. En las líneas de la 3 a la 7 existe un número moderado de correcciones, lo que indica que el robot realiza el proceso con un poco de dificultad sin embargo almacena y despacha los productos sin problemas. En lo que concierne a las líneas 8, 9 y 10 se presentan valores elevados de correcciones, indicando de esta manera que la dificultad de desplazarse es mayor para el robot. En este punto el robot almacena y despacha los productos pero en algunas ocasiones tiende a salirse de la trayectoria planteada.

Por lo tanto, en la Tabla 4.3 se presentan los resultados obtenidos al realizar las pruebas de desplazamiento del robot.

Tabla 4.3: Datos obtenidos en el desplazamiento de Robotino

Pruebas de Desplazamiento General Realizadas	Objetivo Cumplido	Objetivo No Cumplido	Porcentaje de Confiabilidad	Porcentaje de Error
90	75	15	83%	17%

Datos obtenidos en el desplazamiento de Robotino. Elaborado por: Zaira Calvopiña.

Como se muestra en la Tabla 4.3 se realizaron un total de noventa pruebas de desplazamiento del robot por el almacén, de las cuales setenta y cinco desplazamientos se realizaron eficientemente y sin ningún problema y un total de quince desplazamientos no cumplieron con el objetivo planteado, es decir, el robot se salió de la trayectoria deseada. De acuerdo a los datos antes expuestos se obtuvo un porcentaje de confiabilidad del 83% y un porcentaje de error del 17% del proceso general de almacenamiento y despacho de productos.

4.3.2 Tiempos de Almacenamiento y Despacho

En un almacén siempre es necesario conocer el tiempo que se emplea para el almacenaje y despacho de los productos, es por eso que se ha tomado datos del tiempo que Robotino tarda en realizar este proceso, obteniendo los resultados presentados en la Tabla 4.4.

Tabla 4.4: Datos de Tiempo de Almacenamiento y Despacho de Productos

Nº	PRODUCTO	Línea	Nivel	PRUEBA 1	PRUEBA 2	PRUEBA 3
				Tiempo (min:s)	Tiempo (min:s)	Tiempo (min:s)
1	Acetona	1	1	00:25	00:27	00:28
2	Berilio	1	2	00:41	00:40	00:39
3	Fosgeno	1	3	00:56	00:54	00:55
4	Acroleína	2	1	00:34	00:35	00:37
5	Boro	2	2	00:48	00:47	00:46
6	Mercurio	2	3	01:00	01:00	01:00
7	Aluminio	3	1	00:43	00:43	00:43
8	Cadmio	3	2	00:56	00:54	00:54
9	Gasolina	3	3	01:09	01:12	01:08
10	Amianto	4	1	00:49	00:48	00:50
11	Cianuro	4	2	01:01	01:04	01:02
12	Naftalina	4	3	01:14	01:14	01:15
13	Amoníaco	5	1	00:57	00:56	00:58
14	Cromo	5	2	01:06	01:07	01:08
15	Plomo	5	3	01:19	01:21	01:20
16	Anilina	6	1	01:07	01:06	01:09
17	Cloro	6	2	01:15	01:16	01:15
18	Plutonio	6	3	01:26	01:28	01:31
19	Antimonio	7	1	01:11	01:13	01:12
20	Formaldehído	7	2	01:30	01:33	01:31
21	Radón	7	3	01:33	01:35	01:33
22	Arsénico	8	1	01:24	01:25	01:24
23	Endrina	8	2	01:39	01:38	01:39
24	Tolueno	8	3	01:44	01:45	01:45
25	Bario	9	1	01:39	01:38	01:40
26	Etión	9	2	01:45	01:44	01:43
27	Uranio	9	3	01:52	01:51	01:50
28	Benceno	10	1	01:47	01:48	01:51
29	Fenol	10	2	01:55	01:57	01:55
30	Xileno	10	3	02:00	02:01	02:07
TIEMPO TOTAL DEL PROCESO				31:25	31:40	31:41

Resultados del Tiempo de Almacenamiento y Despacho. Elaborado por: Zaira Calvopiña.

Los datos obtenidos en la Tabla 4.4 corresponden al tiempo que empleó el robot tanto en el almacenamiento como en el despacho de cada uno de los productos. De acuerdo a los datos obtenidos se determina que el robot es lento en el proceso, esto debido a que se ha elegido trabajar con una velocidad moderada para evitar que el robot cometa

posibles errores al momento de su desplazamiento. Pese a que la lentitud del robot es una desventaja en el proceso, se contrarresta con la gran ventaja que posee éste evitando riesgos en la salud del ser humano.

CONCLUSIONES

1. Al realizar la investigación y programación de algoritmos de control para la tarjeta de desarrollo Raspberry Pi y para el robot Robotino de la marca FESTO se pudo organizar la operación de un almacén en tres dimensiones identificando productos nocivos para el ser humano mediante códigos QR y movilizándolos a través del robot.
2. Al realizar la caracterización de Robotino se obtuvo información muy importante que permitió compartir datos digitales y analógicos del robot hacia la Raspberry Pi mediante seis pines suficientes para enviar y recibir una combinación de bits para los códigos QR, además se obtuvo información relevante sobre interfaces, protocolos y niveles de voltaje que se debe tener en cuenta para el manejo de los circuitos de control del eje móvil en “z” y la pinza sujetadora.
3. Mediante la interpretación de códigos QR con visión artificial se realizó la identificación de las características de cada una de las sustancias nocivas para el ser humano y a su vez conocer la ubicación exacta que debe ocupar dentro del almacén y así llevar un registro diario del movimiento y la organización de los productos.
4. Se estableció una comunicación bidireccional entre la tarjeta de desarrollo Raspberry Pi y el robot Robotino de FESTO mediante el software Labview y sus librerías correspondientes a Robotino para que el usuario pueda interactuar de manera remota con el proceso monitoreando y adquiriendo información en tiempo real.

5. Al analizar los datos obtenidos correspondientes a la corrección de línea se concluye que mientras mayor es la distancia que recorre Robotino mayor es el número de correcciones de línea, tal es el caso que en las dos primeras líneas no existe ninguna corrección. En las líneas de la 3 a la 7 existe un número aproximado de 1 a 2 correcciones de línea por producto, y en lo que concierne a las líneas 8, 9 y 10 se presentan valores aproximados de 10 a 17 correcciones de línea por producto.
6. Al analizar los datos obtenidos correspondientes a la corrección de posición se concluye que mientras mayor es la distancia que recorre Robotino mayor es el número de correcciones de posición que realiza el robot, tal es el caso que en las dos primeras líneas que recorre, no existe ninguna corrección. En las líneas de la 3 a la 7 existe un número aproximado de 2 correcciones de posición por producto, y en lo que concierne a las líneas 8, 9 y 10 se presentan un valor aproximado de 8 correcciones de posición por producto.
7. Según los datos obtenidos en la Tabla 4.3 se concluye que el porcentaje de confiabilidad del proceso de almacenamiento y despacho de productos es del 83% y el porcentaje de error es del 17% este debido a que el robot abandonó su trayectoria en quince ocasiones de un total de noventa pruebas.
8. Analizando los datos obtenidos en la Tabla 4.4 correspondientes al tiempo que emplea el robot tanto en el almacenamiento como en el despacho de cada uno de los productos se concluye que el robot es lento, siendo esta una desventaja en el proceso que se contrarresta con la gran ventaja que posee éste evitando riesgos en la salud del ser humano.
9. Al realizar las pruebas de funcionamiento se valida el sistema en un ambiente estructurado de treinta localidades, obteniendo resultados favorables y cumpliendo de esta manera con las expectativas propuestas en los objetivos del proyecto de titulación.

RECOMENDACIONES

Se recomienda revisar minuciosamente las hojas de datos de cada uno de los elementos a utilizar en la realización de circuitos de control.

Se debe tener en cuenta que al descargar las librerías de Labview para Robotino estas deben ser de la misma versión de Labview descargado e instalado en el equipo a utilizar.

Con lo que respecta al lenguaje de programación en Python, se debe tener mucho cuidado a la hora de desarrollar algoritmos de control, ya que este lenguaje es muy estricto y no permite errores de tipeo, es mejor utilizar un editor de código que permita eliminar errores al momento de crear un algoritmo de control.

Se recomienda calibrar bien los sensores ópticos para que cada uno de ellos cense de manera óptima la trayectoria a seguir.

Se recomienda disponer de fusibles automotrices de 5 Amperios ya que son de gran utilidad si se produce un corto circuito por error de conexiones en el robot.

Se recomienda disponer de una cámara de al menos 2 Mega pixeles para que la resolución no afecte la interpretación de los datos en la Raspberry Pi.

CITAS BIBLIOGRÁFICAS

- Agencia para Sustancias Tóxicas y el Registro de Enfermedades. (2016). *Resúmenes de Salud Pública - Índice alfabético*. Atlanta, Estados Unidos. Obtenido de https://www.atsdr.cdc.gov/es/phs/es_phsindex.html
- Apache HTTP Server Project. (2018). *Compilar e instalar*. Obtenido de <http://httpd.apache.org/docs/2.4/es/install.html>
- Bliesener, M., Weber, R. C., Karras, U., Zitzmann, D., Kathmann, T., & Schwarzenberger, D. (2011). *Manual de Trabajo, Robotino, FESTO*. Alemania: Festo Didactic GmbH & Co. KG. Obtenido de http://www.festo-didactic.com/ov3/media/customers/1100/548606_leseprobe_es.pdf
- Códigos QR. (2009). *Generador de Códigos QR*. Obtenido de <http://www.codigos-qr.com/generador-de-codigos-qr/>
- Cuji Cuji, C. (2013). *Diseño y construcción de un módulo de objetos para el Robot móvil Robotino de Festo para la Carrera de Ingeniería Electrónica, Sede Quito Campus Sur*. Quito, Ecuador. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/5169/1/UPS-ST000979.pdf>
- ELECTRONILAB. (s.f.). *Módulo LM2596 Convertidor de Voltaje DC-DC Buck 1.25V-35V*. Bogotá, Colombia. Obtenido de <https://electronilab.co/tienda/modulo-lm2596-convertidor-de-voltaje-dc-dc-buck-1-25v-35v/>
- Faro de Vigo. (2012). *Futuros ingenieros y profesionales conocen las innovaciones en robótica de la industria*. Obtenido de <https://www.farodevigo.es/gran-vigo/2012/11/13/futuros-ingenieros-profesionales-conocen-innovaciones-robotica-industria/710971.html>
- García Gadea, D. (2016). *Sistema autónomo y de bajo coste para reconocimiento de códigos QR*. Alicante, España. Obtenido de https://docs.google.com/viewerng/viewer?url=https://rua.ua.es/dspace/bitstream/10045/57290/1/Sistema_autonomo_y_de_bajo_coste_para_reconocimiento_d_Garcia_Gadea_Daniel.pdf

- García, J. C., & Okazaki, S. (2012). *El uso de los códigos QR en España*. España: Mercasa. Obtenido de http://www.mercasa.es/files/multimedios/1342895653_DyC_123_pag_046-062_Andres-Okazaki.pdf
- Gobierno de España, Ministerio de Educación. (2012). *Visión Artificial. Aplicación práctica de la visión artificial en el control de procesos industriales*. España. Obtenido de http://visionartificial.fpcat.cat/wp-content/uploads/UD_1_DIDAC.pdf
- Ibrugor. (2014). *Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve?* Obtenido de <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>
- Logitech. (2011). *Webcam C170*. Obtenido de <http://pcz.mx/pdfp/AC-230663-125.pdf>
- Martínez Rodríguez, A. (2014). *Sistema Automatizado de Reconocimiento y Manipulación de objetos usando visión por computadora y un brazo industrial*. León, Guanajuato, México. Obtenido de <http://biblioteca.cio.mx/tesis/16050.pdf>
- Medina, L. (2013). *Como instalar PHPen Debian*. Obtenido de <https://www.comoinstalarlinux.com/como-instalar-php-en-debian/>
- Mil y un partes. (2014). *Sensor de Campo Magnético Arduino KY-003 (Módulo)*. Bogotá, Colombia. Obtenido de <http://milyunpartes.com/otros-tipos/23-arduino-ky-003-modulo-sensor-de-campo-magnetico.html>
- Palenzuela, J. L. (2016). *Los 10 problemas más comunes en gestión de almacén para las Pymes*. Obtenido de <http://blog.neteris.com/stepforward/blog/puedo-optimizar-procesos-en-mi-almacen-problemas-pymes>
- raspberrypi.org. (s.f.). *Raspberry Pi 3 Model B+*. Reino Unido. Obtenido de <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>

- Schmalz, B. (2015). *Conductor de motor paso a paso Easy Driver*. Estados Unidos. Obtenido de <http://www.schmalzhaus.com/EasyDriver/>
- tecnikro. (s.f.). *Qué es un servomotor y cómo se utiliza?* Quito, Ecuador. Obtenido de <http://programarpicenc.com/articulos/que-es-un-servomotor-y-como-se-utiliza/>
- Universidad Autónoma de Occidente. (2011). *Manual de Manejo Seguro de Productos Químicos*. Cali, Colombia. Obtenido de <https://red.uao.edu.co/bitstream/10614/3035/5/Anexo%2023.%20Manual%20de%20Manejo%20Seguro%20de%20Productos%20Quimicos..pdf>
- Volver, A. (2012). *Raspberry. Una inversión en el futuro de la Ingeniería*. Birchington Road. Weldon, Corby: RS Components Limited. Obtenido de http://etech.designspark.info/ELE_0050_eTech%2010/ELE_0050_eTech_ES/pubData/source/ELE_0050_eTech_ES.pdf
- Weber , R. C., Bellenberg, M., & Schwarzenberger, D. (2010). *Manual, Robotino, FESTO*. Alemania: Festo Didactic GmbH & Co. KG. Obtenido de http://www.festo-didactic.com/ov3/media/customers/1100/544305_robotino_esfr2.pdf

ANEXO I

Algoritmo de control para motor a pasos

```
import RPi.GPIO as gpio
import time
import sys

gpio.setmode(gpio.BCM)

gpio.setup(9, gpio.OUT)          ##salida raspb pin 9 - easydriver(dir)
gpio.setup(10, gpio.OUT)         ##salida raspb pin 10 - easydriver(step)
gpio.setup(18, gpio.IN)          ##entrada raspb pin 18 - señal para subir motor
gpio.setup(23, gpio.IN)         ##entrada raspb pin 23 - señal para bajar motor
gpio.setup(24, gpio.IN)         ##entraba raspb pin 24 - señal para sensor magnetico

gpio.setup(11, gpio.IN,gpio.PUD_DOWN) ##entrada final de carrera abajo
gpio.setup(5, gpio.IN,gpio.PUD_DOWN)  ##entrada final de carrera arriba
gpio.setup(7, gpio.IN)               ##entrada sensor magnetico

def set_stepper_on():
    gpio.output(9, False)
    time.sleep(0.0004)
    gpio.output(9, True)
    time.sleep(0.0004)

infinite_loop = True
steps=0
while (infinite_loop == True):
    gpio.output(9, False)

    steps+=2600

    if gpio.input(18) == True and gpio.input(11) == False:
        set_stepper_on()
        gpio.output(10, False)
        if gpio.input(7) == True:
            position=1

    if gpio.input(23) == True and gpio.input(5) == False:
        set_stepper_on()
        gpio.output(10, True)
        if gpio.input(7) == True:
            position=0

    if gpio.input(24) == True and gpio.input(7) == True:
        set_stepper_on()
        if position ==1:
            gpio.output(10, True)
        else:
            gpio.output(10, False)
```

Algoritmo de control para motor a pasos desarrollado en el software Phytton. Elaborado por: Zaira Calvopiña

ANEXO II

Algoritmo de control para pinza sujetadora

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(12, GPIO.OUT)    ##salida raspb pin 12 - servomotor
GPIO.setup(16, GPIO.IN, GPIO.PUD_UP) ##entrada raspb pin 16 - señal para abrir pinza
GPIO.setup(8, GPIO.IN, GPIO.PUD_UP)  ##entrada raspb pin 8 - señal para cerrar pinza

p = GPIO.PWM(12, 50)
p.start(2.5)
try:
    while True:
        if GPIO.input(16) == True:
            p.ChangeDutyCycle(5)    ##ángulo de apertura de pinza
        if GPIO.input(8) == True:
            p.ChangeDutyCycle(10)   ##ángulo de cierre de pinza
except KeyboardInterrupt:
    p.stop()
    GPIO.cleanup()
```

Algoritmo de control de la pinza sujetadora desarrollado en el software Phyton. Elaborado por: Zaira Calvopiña

ANEXO III

Algoritmo de lectura e interpretación de códigos QR

```
import RPi.GPIO as GPIO
import time
import sys
GPIO.setmode(GPIO.BCM)

GPIO.setup(2, GPIO.OUT) ## GPIO 2 como salida, Trama codigos QR, Bit 1
GPIO.setup(3, GPIO.OUT) ## GPIO 3 como salida, Trama codigos QR, Bit 2
GPIO.setup(4, GPIO.OUT) ## GPIO 4 como salida, Trama codigos QR, Bit 3
GPIO.setup(17, GPIO.OUT) ## GPIO 17 como salida, Trama codigos QR, Bit 4
GPIO.setup(27, GPIO.OUT) ## GPIO 27 como salida, Trama codigos QR, Bit 5
GPIO.setup(22, GPIO.OUT) ## GPIO 22 como salida, Trama codigos QR, Bit 6

def lectura():
    archivo = open("raspi.txt", "r")
    lineal = archivo.readline()
    bina = bin(int(lineal))
    binario = bina[2:len(bina)]

    binario_completo = binario.zfill(6)

    bits = list(binario_completo)

    x=6
    for bit in bits:
        if (x==6):
            # led 6
            if (bit=='1'):
                GPIO.output(22, True) ## Enciendo el 22
            else:
                GPIO.output(22, False) ## Enciendo el 22
        if (x==5):
            # led 5
            if (bit=='1'):
                GPIO.output(27, True) ## Enciendo el 27
            else:
                GPIO.output(27, False) ## Enciendo el 27
        if (x==4):
            # led 4
            if (bit=='1'):
                GPIO.output(17, True) ## Enciendo el 17
            else:
                GPIO.output(17, False) ## Enciendo el 17
        if (x==3):
            # led 3
            if (bit=='1'):
                GPIO.output(4, True) ## Enciendo el 4
            else:
                GPIO.output(4, False) ## Enciendo el 4
        if (x==2):
            # led 2
            if (bit=='1'):
                GPIO.output(3, True) ## Enciendo el 3
            else:
                GPIO.output(3, False) ## Enciendo el 3
        if (x==1):
            # led 1
            if (bit=='1'):
                GPIO.output(2, True) ## Enciendo el 2
            else:
                GPIO.output(2, False) ## Enciendo el 2

        x=x-1

    while(1):
        lectura()
        time.sleep(1)
        GPIO.cleanup() ## Hago una limpieza de los GPIO
```

Algoritmo de lectura e interpretación de códigos QR desarrollado en el software Phyton. Elaborado por: Zaira Calvopiña.

ANEXO IV

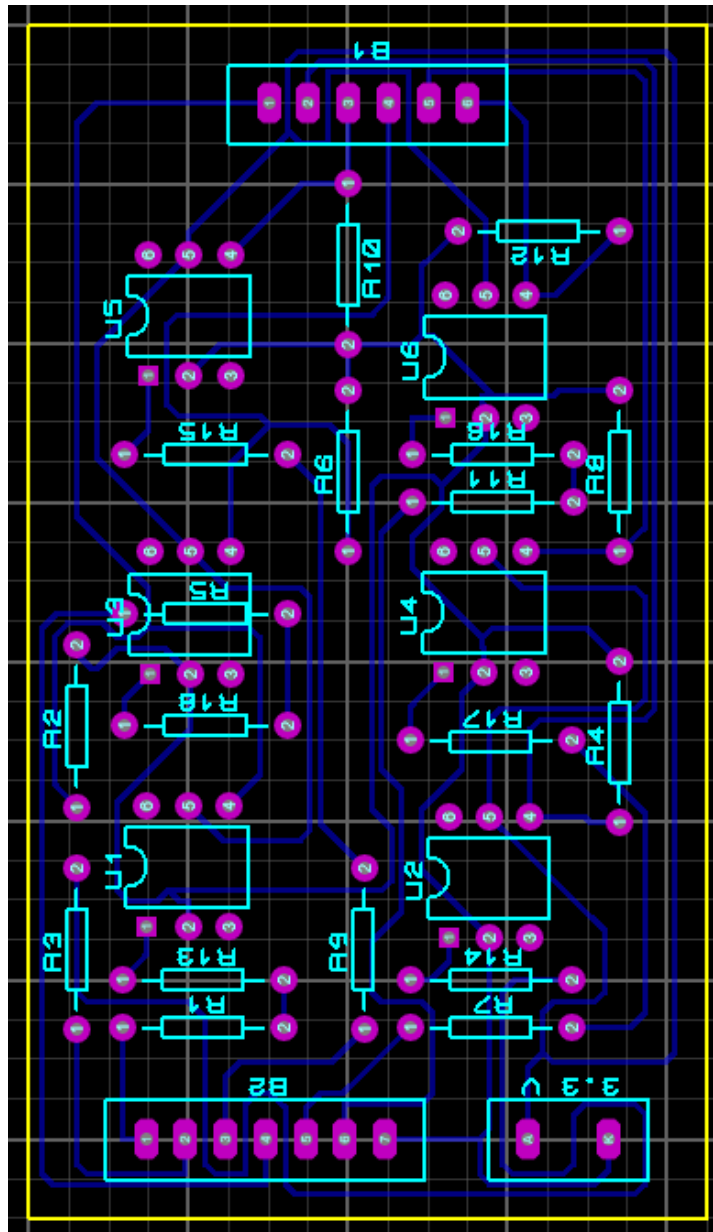
Pines de la Placa Raspberry Pi 3

PINES RASPBERRY PI 3					
Alimentación (+) opto acopladores	3.3V PWR	1	2	5V PWR	Alimentación (+) Finales de carrera
Pin código QR	GPIO 2	3	4	5V PWR	Alimentación (+) ventilador
Pin código QR	GPIO 3	5	6	GND	Alimentación (-) ventilador
Pin código QR	GPIO 4	7	8	UARTD TX	
(-) código QR	GND	9	10	UARTD RX	
Pin código QR	GPIO 17	11	12	GPIO 18	Señal – subir motor
Pin código QR	GPIO 27	13	14	GND	
Pin código QR	GPIO 22	15	16	GPIO 23	Señal – bajar motor
	3.3V PWR	17	18	GPIO 24	Señal – sensor magnético
Easy Driver (Dir)	GPIO 10	19	20	GND	
Easy Driver (Step)	GPIO 9	21	22	GPIO 25	
F. Carrera inferior	GPIO 11	23	24	GPIO 8	Señal – cerrar pinza
	GND	25	26	GPIO 7	Sensor magnético
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
F. Carrera superior	GPIO 6	31	32	GPIO 12	Señal – servomotor
	GPIO 13	33	34	GND	
	GPIO 19	35	36	GPIO 16	Señal – abrir pinza
	GPIO 26	37	38	GPIO 20	
Alimentación (-) opto acopladores	GND	39	40	GPIO 21	

Pines de la placa Raspberry Pi 3 utilizados en el proyecto. Elaborado por: Zaira Calvopiña.

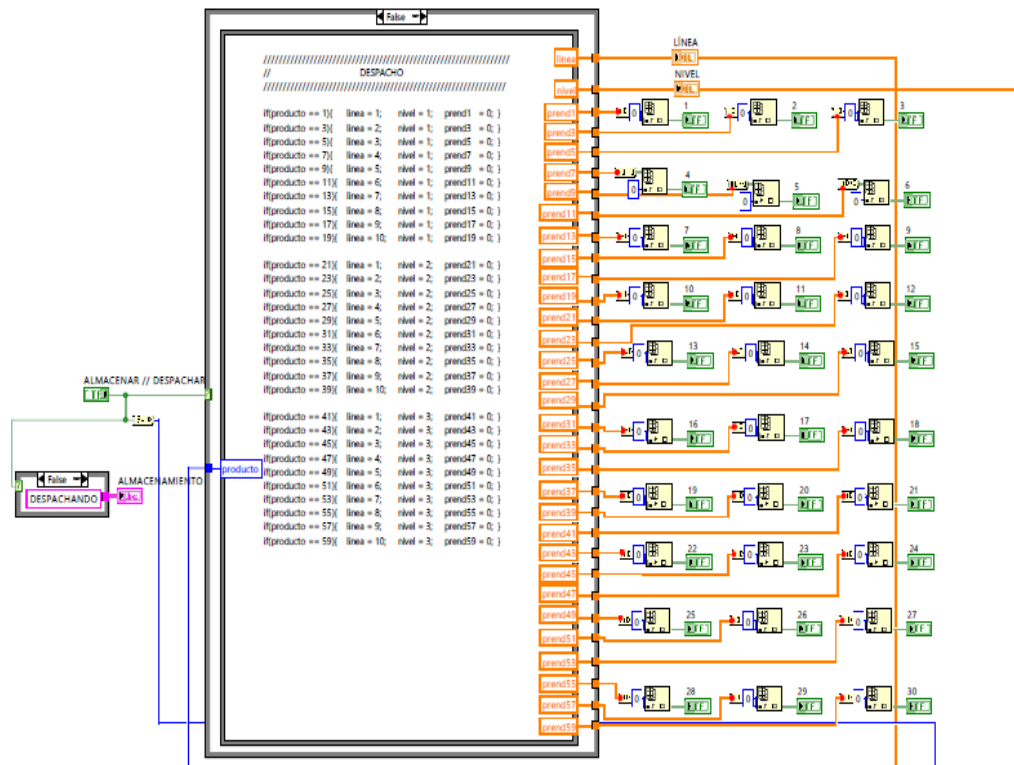
ANEXO V

Diseño de la placa de opto acopladores en PCB



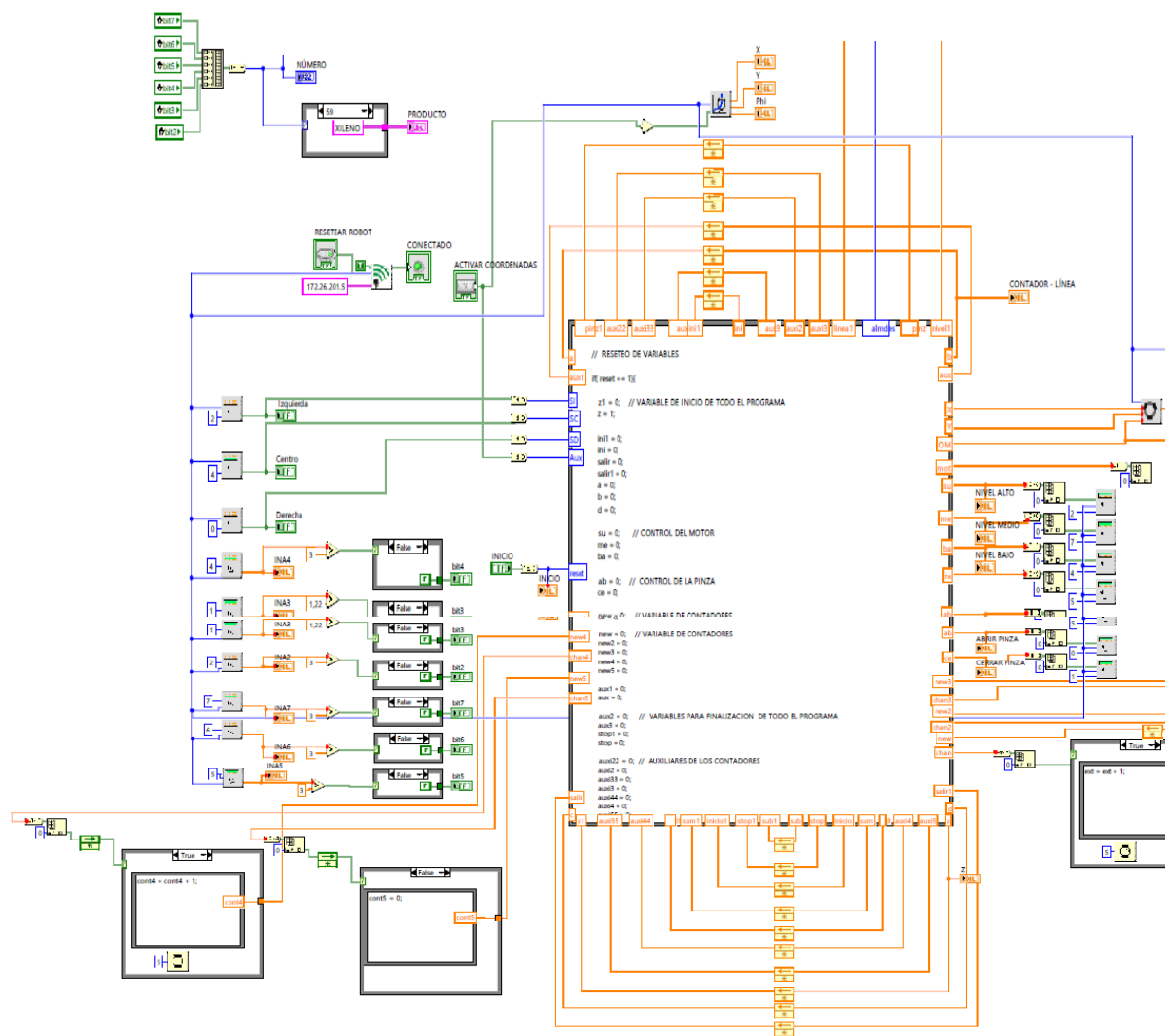
Diseño de la placa de opto acopladores en PCB Layout. Elaborado por: Zaira Calvopiña.

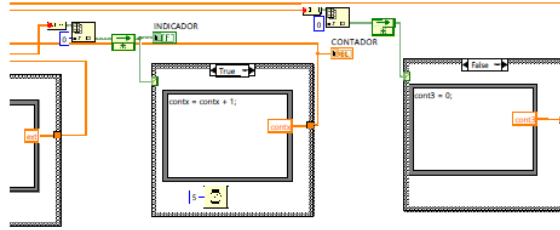
Algoritmo de Control General desarrollado en Labview 2015



Algoritmo de control general desarrollado en LabView 2015 para el sistema de almacenamiento y despacho de productos. Elaborado por: Zaira Calvopiña.

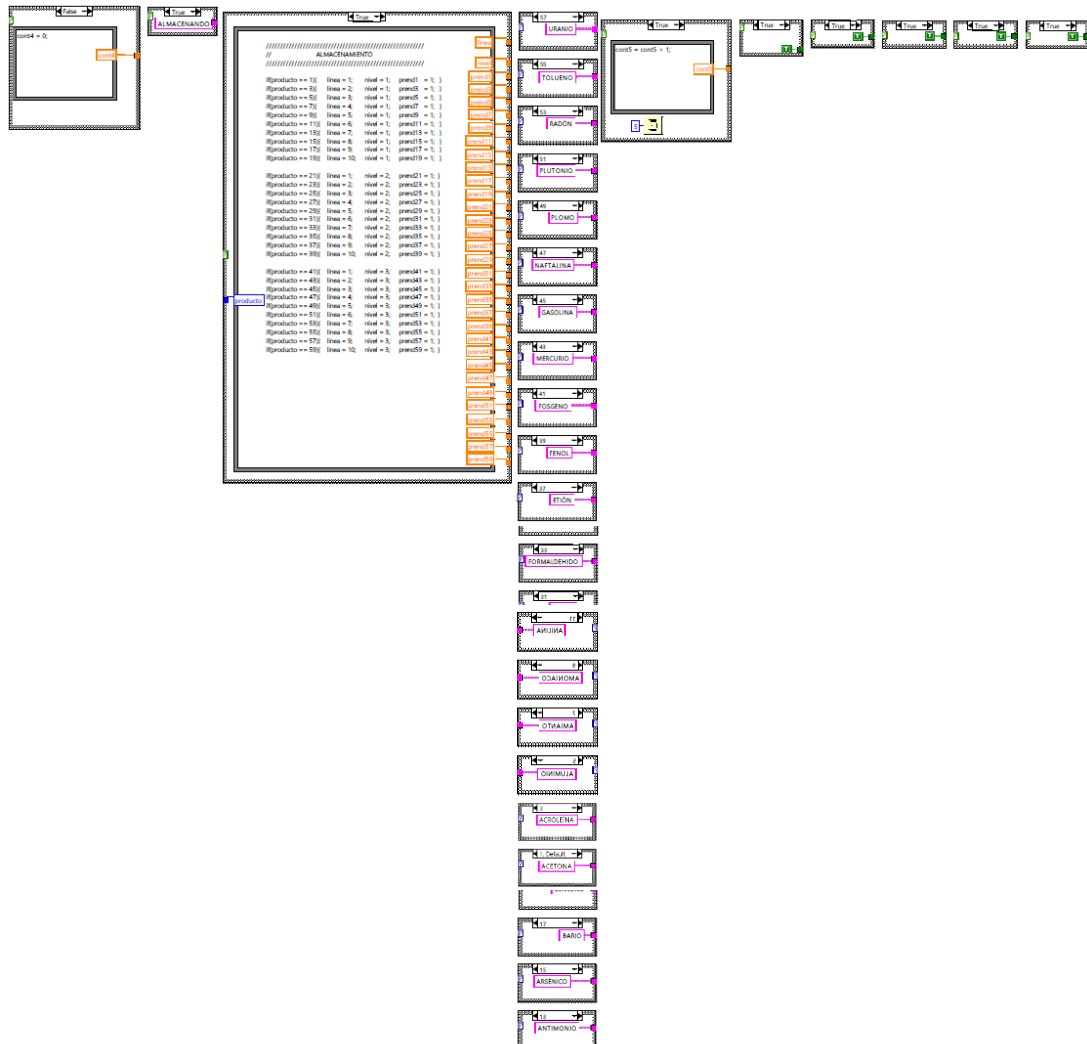
Algoritmo de control general desarrollado en Labview 2015





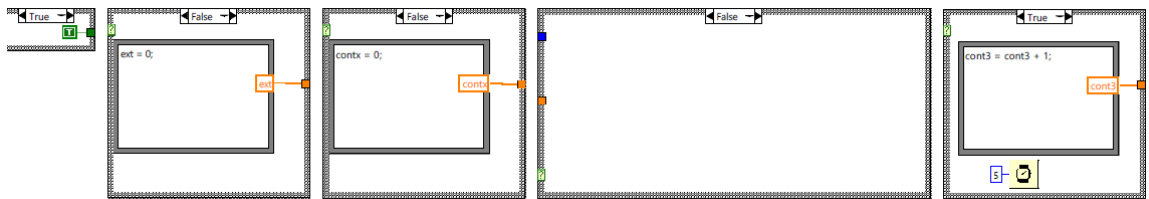
Algoritmo de control general desarrollado en LabView 2015 para el sistema de almacenamiento y despacho de productos. Elaborado por: Zaira Calvopiña.

Algoritmo de control general desarrollado en Labview 2015



Algoritmo de control general desarrollado en LabView 2015 para el sistema de almacenamiento y despacho de productos. Elaborado por: Zaira Calvopiña.

Algoritmo de control general desarrollado en Labview 2015



Algoritmo de control general desarrollado en LabView 2015 para el sistema de almacenamiento y despacho de productos. Elaborado por: Zaira Calvopiña.

